SRH Categorical and Dichotomized

Christine Lucille Kuryla

2025-01-09

```r
data_gss <- read_csv(here("data/extracted_gss_variables.csv")) %>%
  filter(cohort != 9999) %>%
  filter(!(is.na(health))) %>%
#  mutate(south = if_else(region %in% c(4, 5, 6, 7, 8), "South", "Not South")) %>%
#  mutate(south = as_factor(south)) %>%
#  mutate(region = as_factor(region)) %>%
 # mutate(wrkstat = as_factor(wrkstat)) %>%
#  mutate(hispanic = as_factor(hispanic)) %>%
 # mutate(marital = as_factor(marital)) %>%
#  mutate(partyid = as_factor(partyid)) %>%
  mutate(health = 5 - health)  %>%  # reverse the coding so it's more intuitive (higher number for exce
  mutate(happy = 4 - happy) %>% # same
  mutate(life = 4 - life) %>% # reverse again, these variables tend to be unintuitively ordered!!!
  mutate(satfin = 4 - satfin) %>% # same again! %>%
 mutate(
    age_group = cut(
      age,
      breaks = c(17, 29, 39, 49, 59, 69, Inf),
      labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
      right = TRUE
    )
  )
```

```
## Rows: 72390 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (13): year, cohort, age, health, sex, happy, life, educ, polviews, class...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
table(data_gss$age_group)
```

```
##
## 18-29 30-39 40-49 50-59 60-69   70+
## 10922 11263  9743  8435  7206  6892
```

```r
# more categories

data_gss <- data_gss %>%
    mutate(
    health_cat = factor(health,
                        levels = 1:4,
                        labels = c("Poor", "Fair", "Good", "Excellent")),
```

```r
  period_cut_6 = as.factor(cut(data_gss$year, 6)),
  period_cut_10 = as.factor(cut(data_gss$year, 10)),
  period_cut_12 = as.factor(cut(data_gss$year, 12)),
  period_groups = as.factor(cut(data_gss$year, 12)),
  period_10yr = as.factor(
                        cut(
                        year,
                        breaks = c(1971, 1982, 1990, 1998, 2006, 2014, Inf),
                        labels = c("1972-1982", "1982-1990", "1990-1998",
                                   "1998-2006", "2006-2014", "2014-2022"),
                        right = TRUE
                        )
                      ),
  period_decade = as.factor(
                        cut(
                        year,
                        breaks = c(1971, 1979, 1989, 1999, 2009, 2019, Inf),
                        labels = c("1972-1979", "1980-1989", "1990-1999",
                                   "2000-2009", "2010-2019", "2020-2024"),
                        right = TRUE
                        )
                      ),
  age_group = as.factor(
                        cut(
                        age,
                        breaks = c(17, 29, 39, 49, 59, 69, Inf),
                        labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                        right = TRUE
                      )),
  age_groups = as.factor(
                        cut(
                        age,
                        breaks = c(17, 29, 39, 49, 59, 69, Inf),
                        labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                        right = TRUE
                      )),
  age_group_small = as.factor(
                        cut(
                          age,
                          breaks = c(seq(15, 75, by = 5), Inf),  # Define breaks up to 75 and inclu
                          labels = c("16-20", "21-25", "26-30", "31-35", "36-40", "41-45", "46-50",
                          right = FALSE  # Makes intervals left-closed, i.e., [x, y)
                        )
                      )
                      ,
  generation = factor(
    case_when(
      cohort >= 1901 & cohort <= 1927 ~ "Greatest (1901-1927)",
      cohort >= 1928 & cohort <= 1945 ~ "Silent (1928-1945)",
      cohort >= 1946 & cohort <= 1964 ~ "Boomers (1946-1964)",
      cohort >= 1965 & cohort <= 1980 ~ "Gen X (1965-1980)",
      cohort >= 1981 & cohort <= 1996 ~ "Millennials (1981-1996)",
      cohort >= 1997 & cohort <= 2012 ~ "Gen Z (1997-2012)",
```

```r
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest (1901-1927)",
      "Silent (1928-1945)",
      "Boomers (1946-1964)",
      "Gen X (1965-1980)",
      "Millennials (1981-1996)",
      "Gen Z (1997-2012)"#,
   #   "Other"
    )
  ),
  generation_two_sections = factor(
    case_when(
      generation == "Greatest (1901-1927)" & cohort <= 1914 ~ "Greatest Early (1901-1914)",
      generation == "Greatest (1901-1927)" & cohort > 1914 ~ "Greatest Late (1915-1927)",
      generation == "Silent (1928-1945)" & cohort <= 1936 ~ "Silent Early (1928-1936)",
      generation == "Silent (1928-1945)" & cohort > 1936 ~ "Silent Late (1937-1945)",
      generation == "Boomers (1946-1964)" & cohort <= 1955 ~ "Boomers Early (1946-1955)",
      generation == "Boomers (1946-1964)" & cohort > 1955 ~ "Boomers Late (1956-1964)",
      generation == "Gen X (1965-1980)" & cohort <= 1972 ~ "Gen X Early (1965-1972)",
      generation == "Gen X (1965-1980)" & cohort > 1972 ~ "Gen X Late (1973-1980)",
      generation == "Millennials (1981-1996)" & cohort <= 1988 ~ "Millennials Early (1981-1988)",
      generation == "Millennials (1981-1996)" & cohort > 1988 ~ "Millennials Late (1989-1996)",
      generation == "Gen Z (1997-2012)" & cohort <= 2004 ~ "Gen Z Early (1997-2004)",
      generation == "Gen Z (1997-2012)" & cohort > 2004 ~ "Gen Z Late (2005-2012)",
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest Early (1901-1914)", "Greatest Late (1915-1927)",
      "Silent Early (1928-1936)", "Silent Late (1937-1945)",
      "Boomers Early (1946-1955)", "Boomers Late (1956-1964)",
      "Gen X Early (1965-1972)", "Gen X Late (1973-1980)",
      "Millennials Early (1981-1988)", "Millennials Late (1989-1996)",
      "Gen Z Early (1997-2004)", "Gen Z Late (2005-2012)"#,
   #   "Other"
    )
  ),
  generation_three_sections = factor(
    case_when(
      generation == "Greatest (1901-1927)" & cohort <= 1910 ~ "Greatest Early (1901-1910)",
      generation == "Greatest (1901-1927)" & cohort > 1910 & cohort <= 1918 ~ "Greatest Mid (1911-1918
      generation == "Greatest (1901-1927)" & cohort > 1918 ~ "Greatest Late (1919-1927)",
      generation == "Silent (1928-1945)" & cohort <= 1934 ~ "Silent Early (1928-1934)",
      generation == "Silent (1928-1945)" & cohort > 1934 & cohort <= 1940 ~ "Silent Mid (1935-1940)",
      generation == "Silent (1928-1945)" & cohort > 1940 ~ "Silent Late (1941-1945)",
      generation == "Boomers (1946-1964)" & cohort <= 1951 ~ "Boomers Early (1946-1951)",
      generation == "Boomers (1946-1964)" & cohort > 1951 & cohort <= 1958 ~ "Boomers Mid (1952-1958)
      generation == "Boomers (1946-1964)" & cohort > 1958 ~ "Boomers Late (1959-1964)",
      generation == "Gen X (1965-1980)" & cohort <= 1970 ~ "Gen X Early (1965-1970)",
      generation == "Gen X (1965-1980)" & cohort > 1970 & cohort <= 1976 ~ "Gen X Mid (1971-1976)",
      generation == "Gen X (1965-1980)" & cohort > 1976 ~ "Gen X Late (1977-1980)",
      generation == "Millennials (1981-1996)" & cohort <= 1986 ~ "Millennials Early (1981-1986)",
```

```r
      generation == "Millennials (1981-1996)" & cohort > 1986 & cohort <= 1992 ~ "Millennials Mid (198
      generation == "Millennials (1981-1996)" & cohort > 1992 ~ "Millennials Late / Gen Z (1993-2004)
  #    generation == "Gen Z (1997-2012)" & cohort <= 2002 ~ "Gen Z Early (1997-2002)",
  #    generation == "Gen Z (1997-2012)" & cohort > 2002 & cohort <= 2008 ~ "Gen Z Mid (2003-2008)",
  #    generation == "Gen Z (1997-2012)" & cohort > 2008 ~ "Gen Z Late (2009-2012)",
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest Early (1901-1910)", "Greatest Mid (1911-1918)", "Greatest Late (1919-1927)",
      "Silent Early (1928-1934)", "Silent Mid (1935-1940)", "Silent Late (1941-1945)",
      "Boomers Early (1946-1951)", "Boomers Mid (1952-1958)", "Boomers Late (1959-1964)",
      "Gen X Early (1965-1970)", "Gen X Mid (1971-1976)", "Gen X Late (1977-1980)",
      "Millennials Early (1981-1986)", "Millennials Mid (1987-1992)",
      "Millennials Late / Gen Z (1993-2004)"
      #"Millennials Late (1993-1996)",
    #  "Gen Z Early (1997-2002)", "Gen Z Mid (2003-2008)", "Gen Z Late (2009-2012)" #,
    #  "Other"
    )
  )
) %>%
  mutate(age_group = as_factor(age_group))
```

Dichotomize

```r
# Define the full set of categories
all_levels <- c("Excellent", "Good", "Fair", "Poor")

# Use the combinatorial 'combn' function to get all subsets
all_subsets <- map(1:length(all_levels), ~combn(all_levels, m = .x, simplify = FALSE)) %>%
  # Flatten the list of lists
  flatten()

# Filter out the empty set (already not returned by combn) and the full set
# Keep everything else: these are the potential "Group 1"s
valid_subsets <- all_subsets %>%
  discard(~length(.x) == 4)  # remove subset with all 4 categories

# delete the weird ones

valid_subsets <- valid_subsets[c(1:5, 11)]

# Create dichotomized variables

# A helper function to turn a subset (S) into a meaningful column name
make_dicho_name <- function(S) {
  # Example name: "Ex_Go" vs "Fa_Po" for c("Excellent","Good"), etc.
  # We'll join group names with underscores and separate them with "v".
  group1_name <- str_c(S, collapse = "_")
  # We find the complement
  group2 <- setdiff(all_levels, S)
  group2_name <- str_c(group2, collapse = "_")
  # Combine them
```

```r
  out_name <- str_c("health_dicho_", group1_name, "_v_", group2_name)
  # For cleanliness, ensure no weird characters/spaces:
  out_name <- str_replace_all(out_name, "\\s+", "")
  out_name
}

data_gss_dichotomized <- data_gss

for (subset_i in valid_subsets) {

  new_col_name <- make_dicho_name(subset_i)

  # Dichotomize: if health_cat is in 'subset_i', call it "Group1", else "Group2"
  data_gss_dichotomized <- data_gss_dichotomized %>%
    mutate(
      !!sym(new_col_name) := if_else(health_cat %in% subset_i,
                                     1, 0) #"Group1",
                                     #"Group2")
    )
}

data_gss <- data_gss_dichotomized

dichotomized_var <- colnames(data_gss[(length(data_gss) - length(valid_subsets)+1):(length(data_gss))])

important_dichotomized_var <- dichotomized_var[c(1, 5, 6)]
```

Survey object

```r
# Create a survey design object using wtssall for multi-year analysis
gss_svy <- data_gss %>%
  as_survey_design(
    ids = 1,           # PSU identifiers (use 1 if not available)
    weights = wtsscomp  # wtssall pre 2018, wtsscomp combined //Use 'wtss' for single-year analysis
  )
```

Functions for the analyses

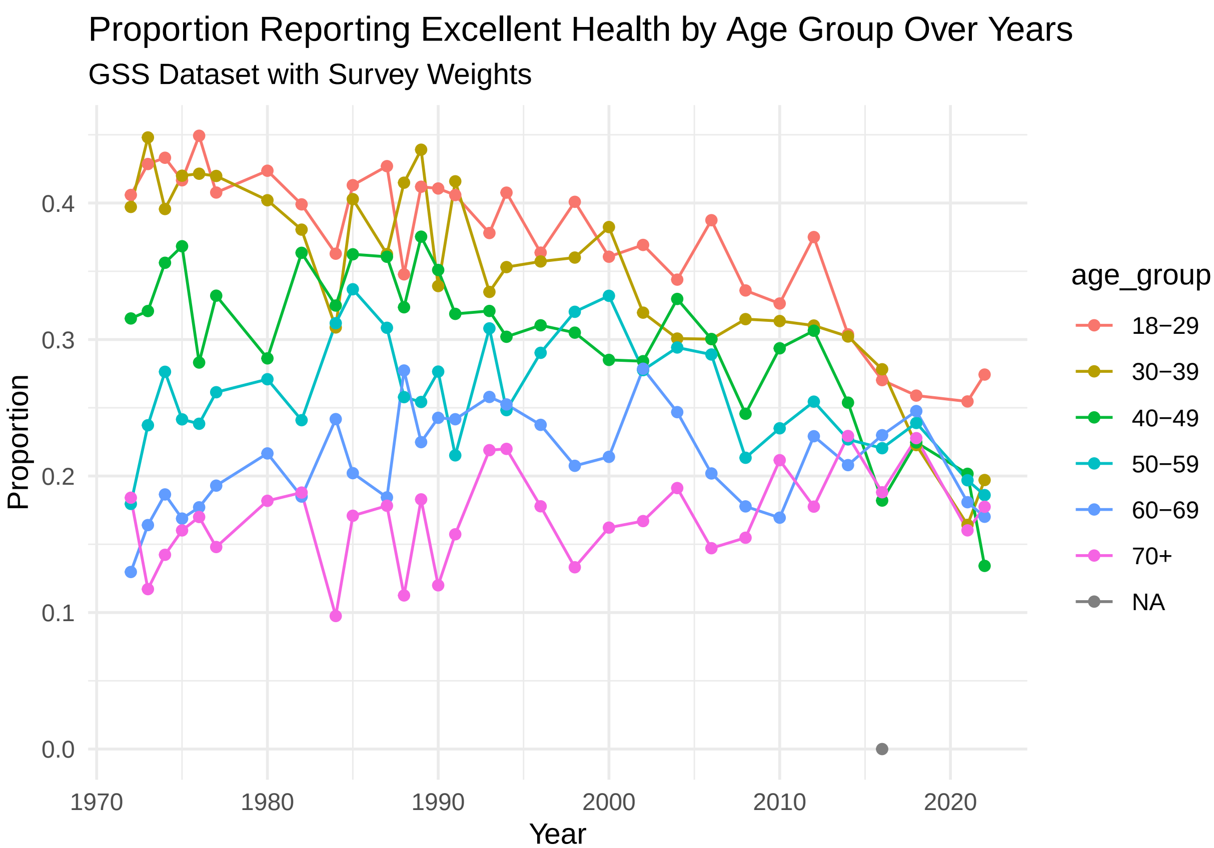
Function to create the first figure

```r
library(survey)
library(srvyr)
library(dplyr)
library(ggplot2)

#' Plot Weighted Proportion of a Dichotomous Variable
#'
#' @param data A srvyr survey object (e.g., as returned by as_survey()).
#' @param outcome A 0/1 variable representing the dichotomous outcome.
#' @param group_var The variable to group by (e.g., age group).
#' @param year_var The variable representing time (e.g., year).
#' @param title Plot title.
```

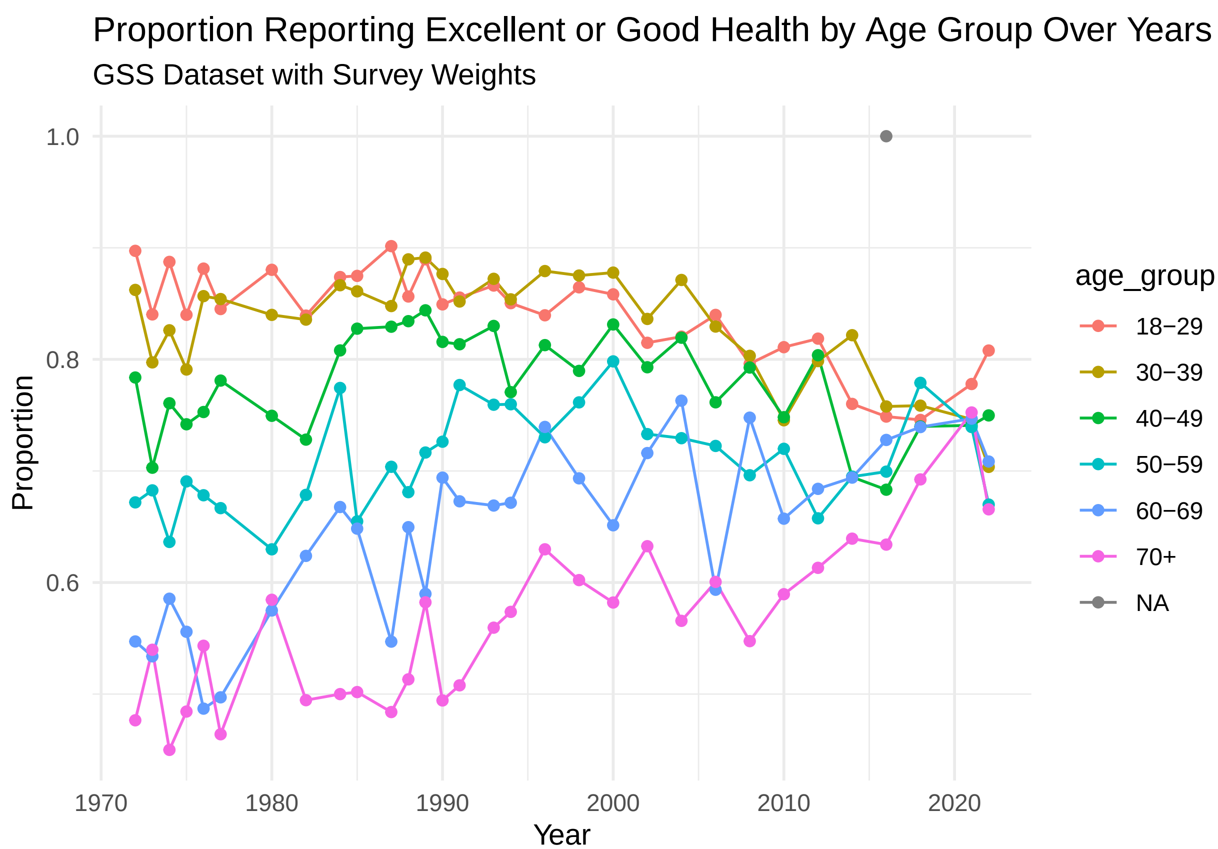
```r
#' @param subtitle Plot subtitle.
#'
#' @return A ggplot object.
#'
#' @examples
#' plot_weighted_proportion(gss_svy, outcome = health_binary,
#'                          group_var = age_group, year_var = year,
#'                          title = "Weighted Proportion by Age Group and Year")
plot_weighted_proportion <- function(data,
                                     outcome,
                                     group_var,
                                     year_var,
                                     title = "Weighted Proportion Over Time",
                                     subtitle = "Survey Data") {
  # Convert unquoted input to symbols for tidy evaluation
  outcome_sym   <- rlang::ensym(outcome)
  group_var_sym <- rlang::ensym(group_var)
  year_var_sym  <- rlang::ensym(year_var)

  # Compute weighted proportion by group and year
  df_summary <- data %>%
    group_by(!!group_var_sym, !!year_var_sym) %>%
    summarise(
      # The mean of a 0/1 variable is the proportion = 1
      prop = survey_mean(!!outcome_sym, na.rm = TRUE),
      .groups = "drop"  # drop last grouping for a cleaner final dataset
    )

  # Create the plot
  p <- df_summary %>%
    ggplot(aes(x = !!year_var_sym, y = prop, color = !!group_var_sym)) +
    geom_line() +
    geom_point() +
    labs(
      title = title,
      subtitle = subtitle,
    #  x = rlang::as_label(year_var_sym),
      x = "Year",
      y = "Proportion",
    #  y = paste0("Proportion of ", rlang::as_label(outcome_sym), " = 1"),
      color = rlang::as_label(group_var_sym)
    ) +
    theme_minimal()

  return(p)
}
```

```r
# "health_dicho_Excellent_v_Good_Fair_Poor"
# "health_dicho_Excellent_Good_v_Fair_Poor"
# "health_dicho_Excellent_Good_Fair_v_Poor"

p_prop_gss_e_vs_gfp <- plot_weighted_proportion(
  data = gss_svy,
  outcome = health_dicho_Excellent_v_Good_Fair_Poor,
```

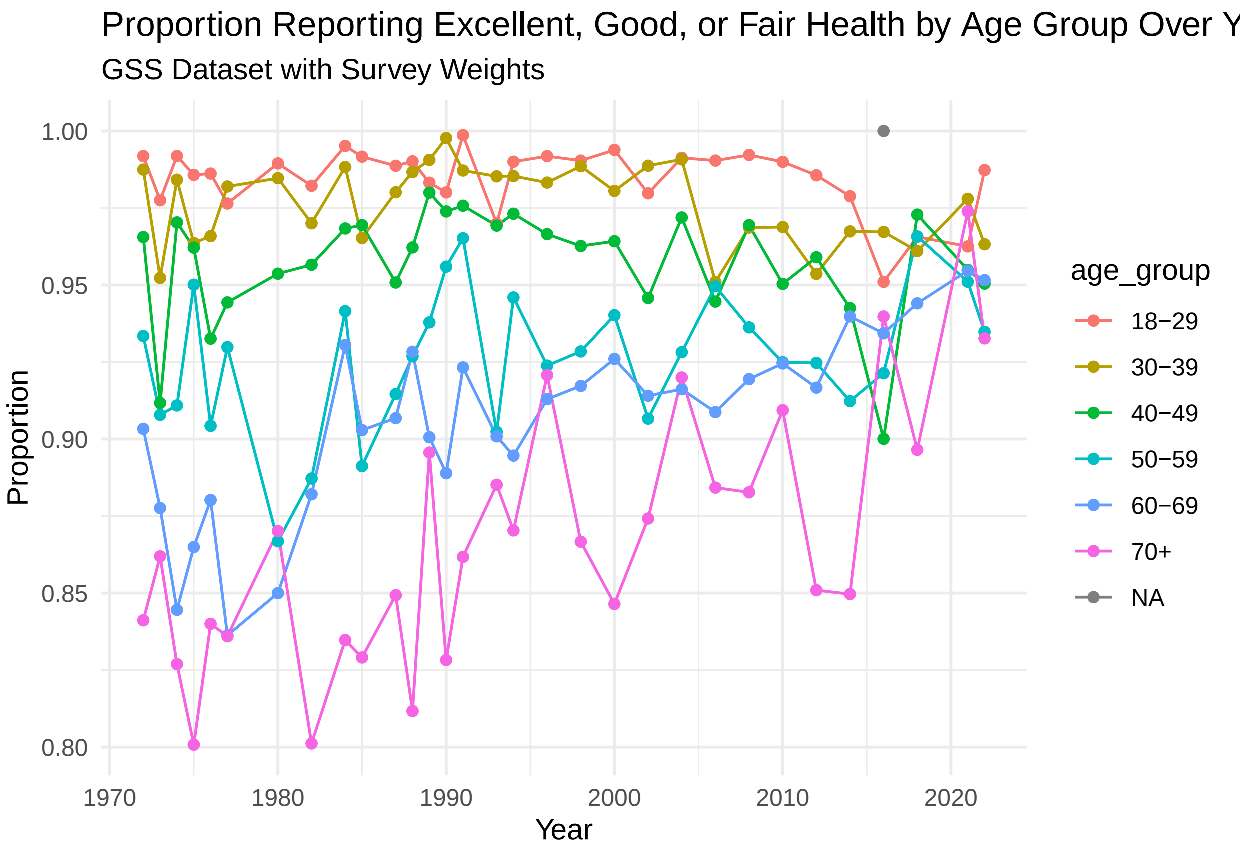
```
  group_var = age_group,
  year_var = year,
  title = "Proportion Reporting Excellent Health by Age Group Over Years",
  subtitle = "GSS Dataset with Survey Weights"
)

p_prop_gss_e_vs_gfp
```




```
p_prop_gss_eg_vs_fp <- plot_weighted_proportion(
  data = gss_svy,
  outcome = health_dicho_Excellent_Good_v_Fair_Poor,
  group_var = age_group,
  year_var = year,
  title = "Proportion Reporting Excellent or Good Health by Age Group Over Years",
  subtitle = "GSS Dataset with Survey Weights"
)
p_prop_gss_eg_vs_fp
```




```r
p_prop_gss_egf_vs_p <- plot_weighted_proportion(
  data = gss_svy,
  outcome = health_dicho_Excellent_Good_Fair_v_Poor,
  group_var = age_group,
  year_var = year,
  title = "Proportion Reporting Excellent, Good, or Fair Health by Age Group Over Years",
  subtitle = "GSS Dataset with Survey Weights"
)

p_prop_gss_egf_vs_p
```



Analyzing Coefficients

```
library(survey)
library(srvyr)
library(dplyr)
library(ggplot2)
library(broom)       # for tidy()
library(knitr)       # for kable() if desired
library(purrr)

#' Weighted Regression of Outcome on a Predictor by Year
#'
#' @param survey_data A srvyr survey design object (e.g., gss_svy).
#' @param outcome The dependent (outcome) variable (e.g., 'health').
#' @param predictor The predictor variable (e.g., 'age').
#' @param year_var The variable that indicates the year (e.g., 'year').
#' @param extra_covars An optional character vector of additional covariates
#'                     (default = NULL).
#' @param alpha The confidence level for conf. intervals (e.g., 0.05 for 95%).
#'
#' @return A list containing:
#'   \item{coefficient_df}{data frame of the regression coefficient for each year.}
#'   \item{coef_plot}{ggplot object of coefficient estimates over time.}
#'   \item{coef_trend_plot}{ggplot object of coefficient estimates with linear fit over time.}
#'   \item{lm_over_time_summary}{lm() summary of how the coefficient changes over time.}
#'
```

```r
#' @details
#' 1. Groups the data by year,
#' 2. Fits a survey-weighted GLM for each group using `svyglm`,
#' 3. Extracts the coefficient of the `predictor` variable,
#' 4. Creates two plots: (1) coefficient vs. year with CIs; (2) the same plus a linear fit over time.
#'
#' @examples
#' # Suppose gss_svy is your survey design object
#' # Weighted regression of health on age for each year:
#' # result_list <- weighted_regressions_by_year(
#' #   survey_data = gss_svy,
#' #   outcome = "health",
#' #   predictor = "age",
#' #   year_var = "year"
#' # )
#' #
#' # result_list$coefficient_df
#' # result_list$coef_plot
#' # result_list$coef_trend_plot
#' # result_list$lm_over_time_summary
weighted_regressions_by_year <- function(survey_data,
                                         outcome,
                                         predictor,
                                         year_var,
                                         extra_covars = NULL,
                                         alpha = 0.05) {

  # --- 1. Build formula for the model ---
  # outcome ~ predictor (+ optional covariates)
  if (!is.null(extra_covars)) {
    rhs <- paste(c(predictor, extra_covars), collapse = " + ")
  } else {
    rhs <- predictor
  }
  formula_str <- paste(outcome, "~", rhs)
  model_formula <- as.formula(formula_str)

  # --- 2. Group by 'year' and fit the models ---
  #   Using group_map_dfr to run one model per 'year'
  coefficient_df <- survey_data %>%
    group_by(.data[[year_var]]) %>%
    group_map_dfr(~ {
      # Fit the weighted linear model
      model <- survey::svyglm(model_formula, design = .x)

      # Tidy up, with conf.int = TRUE for CIs
      # conf.level = 1 - alpha (e.g., 95%)
      model_tidy <- broom::tidy(model, conf.int = TRUE, conf.level = 1 - alpha)

      # Filter for the term == predictor (or if predictor has multiple terms, adjust accordingly)
      filter(model_tidy, term == predictor) %>%
        mutate(
          year = unique(.x[[year_var]])  # store the year for clarity
```

```r
      )
    }) %>%
    ungroup() %>%
    # Select/rename columns for clarity
    select(year,
           term,
           estimate,
           std.error,
           statistic,
           p.value,
           conf.low,
           conf.high)

# --- 3. Plot the coefficients over time ---
coef_plot <- ggplot(coefficient_df, aes(x = year, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                width = 0.2, position = position_dodge(0.05)) +
  labs(
    title = paste("Weighted Coefficients of", predictor, "on", outcome, "by Year"),
    subtitle = paste("Survey data, ~", predictor, if (!is.null(extra_covars)) paste("+", paste(extra_
    x = "Year",
    y = paste0("Coefficient (", predictor, ")")
  ) +
  geom_hline(yintercept = 0,
  color = "maroon", linetype = "dashed", size = 1,      # thickness of the line
  alpha = 0.9    # transparency (light red)
  ) +
  theme_minimal()

# --- 4. Fit a linear model of coefficient ~ year (trend over time) ---
# Convert 'year' to numeric if it's not already
coefficient_df$year <- as.numeric(as.character(coefficient_df$year))
coef_trend_lm <- lm(estimate ~ year, data = coefficient_df)
lm_over_time_summary <- summary(coef_trend_lm)

# Build a second plot with a regression line over time
coef_trend_plot <- ggplot(coefficient_df, aes(x = year, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high),
                width = 0.2, position = position_dodge(0.05)) +
  geom_smooth(method = "lm", se = TRUE, alpha = 0.3, color = "blue") +
  labs(
    title = paste("Trend Over Time for Coefficient of", predictor),
    subtitle = "With linear fit of the coefficient ~ year",
    x = "Year",
    y = paste0("Coefficient (", predictor, ")")
  ) +
  geom_hline(yintercept = 0,
  color = "maroon", linetype = "dashed", size = 1,      # thickness of the line
  alpha = 0.9    # transparency (light red)
  ) +
  theme_minimal()
```

```r
  # --- 5. Return a list of outputs ---
  return(list(
    coefficient_df = coefficient_df,
    coef_plot = coef_plot,
    coef_trend_plot = coef_trend_plot,
    lm_over_time_summary = lm_over_time_summary
  ))
}
```

Apply function

```r
# health_dicho_Excellent_v_Good_Fair_Poor

# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = gss_svy,   # your sruyr design object
  outcome     = "health_dicho_Excellent_v_Good_Fair_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent v Good/Fair/Poor (GSS)")
```

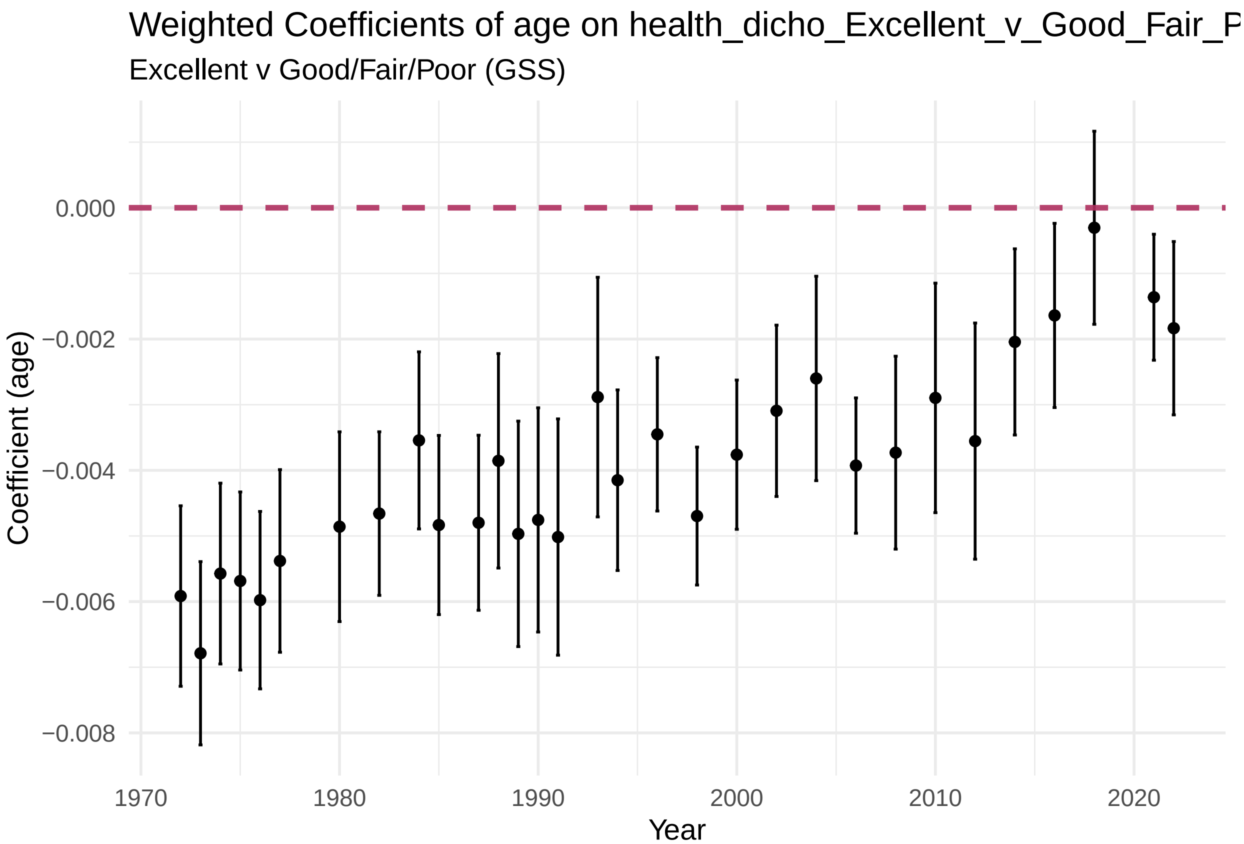
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1972	age	-0.0059152	0.0007009	-8.4392093	0.0000000	-0.0072900	-0.0045404
1973	age	-0.0067870	0.0007117	-9.5370297	0.0000000	-0.0081830	-0.0053911
1974	age	-0.0055727	0.0007021	-7.9372130	0.0000000	-0.0069499	-0.0041955
1975	age	-0.0056857	0.0006915	-8.2225120	0.0000000	-0.0070421	-0.0043294
1976	age	-0.0059778	0.0006890	-8.6759722	0.0000000	-0.0073294	-0.0046263
1977	age	-0.0053805	0.0007091	-7.5877666	0.0000000	-0.0067715	-0.0039896
1980	age	-0.0048587	0.0007367	-6.5951401	0.0000000	-0.0063039	-0.0034136
1982	age	-0.0046588	0.0006353	-7.3337964	0.0000000	-0.0059047	-0.0034129
1984	age	-0.0035438	0.0006875	-5.1542921	0.0000003	-0.0048925	-0.0021951
1985	age	-0.0048332	0.0006958	-6.9462653	0.0000000	-0.0061980	-0.0034684
1987	age	-0.0047988	0.0006798	-7.0591772	0.0000000	-0.0061321	-0.0034655
1988	age	-0.0038551	0.0008323	-4.6315594	0.0000041	-0.0054885	-0.0022217
1989	age	-0.0049680	0.0008752	-5.6766043	0.0000000	-0.0066853	-0.0032506
1990	age	-0.0047555	0.0008701	-5.4655709	0.0000001	-0.0064631	-0.0030479
1991	age	-0.0050161	0.0009172	-5.4688538	0.0000001	-0.0068161	-0.0032162
1993	age	-0.0028843	0.0009301	-3.1009484	0.0019797	-0.0047094	-0.0010592
1994	age	-0.0041504	0.0007014	-5.9171433	0.0000000	-0.0055260	-0.0027748
1996	age	-0.0034518	0.0005957	-5.7946037	0.0000000	-0.0046200	-0.0022837
1998	age	-0.0046966	0.0005358	-8.7659329	0.0000000	-0.0057471	-0.0036460
2000	age	-0.0037616	0.0005797	-6.4892222	0.0000000	-0.0048983	-0.0026248
2002	age	-0.0030932	0.0006653	-4.6493036	0.0000036	-0.0043980	-0.0017884
2004	age	-0.0026003	0.0007936	-3.2764069	0.0010781	-0.0041572	-0.0010434

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2006	age	-0.0039273	0.0005257	-7.4710325	0.0000000	-0.0049579	-0.0028966
2008	age	-0.0037310	0.0007491	-4.9806788	0.0000007	-0.0052005	-0.0022615
2010	age	-0.0028967	0.0008911	-3.2507397	0.0011812	-0.0046449	-0.0011485
2012	age	-0.0035547	0.0009170	-3.8763341	0.0001113	-0.0053538	-0.0017557
2014	age	-0.0020437	0.0007230	-2.8268373	0.0047560	-0.0034617	-0.0006257
2016	age	-0.0016403	0.0007154	-2.2927658	0.0219718	-0.0030434	-0.0002372
2018	age	-0.0003045	0.0007498	-0.4061688	0.6846741	-0.0017753	0.0011662
2021	age	-0.0013629	0.0004894	-2.7848274	0.0053829	-0.0023223	-0.0004034
2022	age	-0.0018348	0.0006733	-2.7250236	0.0064633	-0.0031549	-0.0005146

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

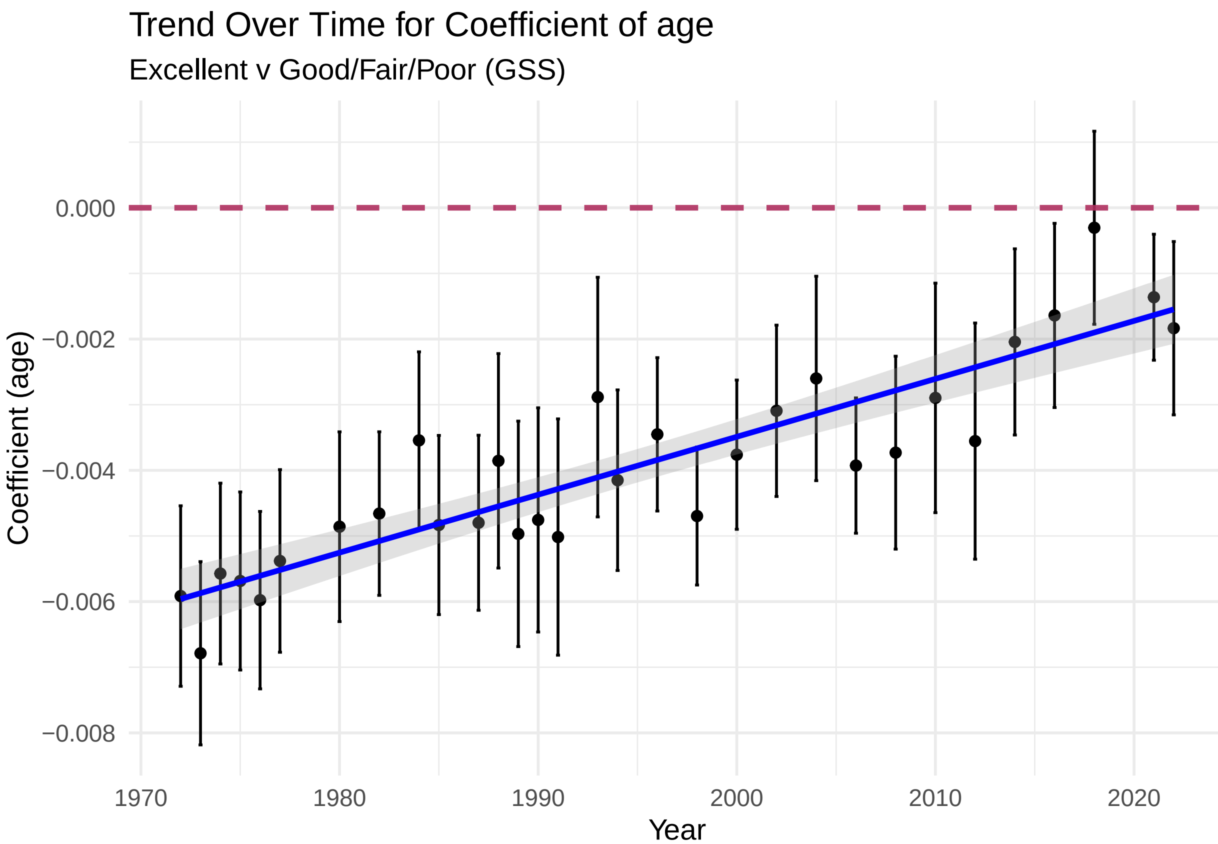
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.125e-03 -3.774e-04  9.530e-06  3.927e-04  1.596e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.800e-01  1.646e-02  -10.94 8.35e-12 ***
## year         8.826e-05  8.251e-06   10.70 1.40e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0006929 on 29 degrees of freedom
## Multiple R-squared:  0.7978, Adjusted R-squared:  0.7908
## F-statistic: 114.4 on 1 and 29 DF,  p-value: 1.404e-11
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent v Good/Fair/Poor (GSS)"))
```




```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent v Good/Fair/Poor (GSS)"))
```

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




```r
p_coeff_gss_e_vs_gfp <- result_list$coef_trend_plot + labs(subtitle = "Excellent v Good/Fair/Poor (GSS)
```

```r
# health_dicho_Excellent_Good_v_Fair_Poor

# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = gss_svy,    # your srvyr design object
  outcome     = "health_dicho_Excellent_Good_v_Fair_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent/Good v Fair/Poor (GSS)")
```

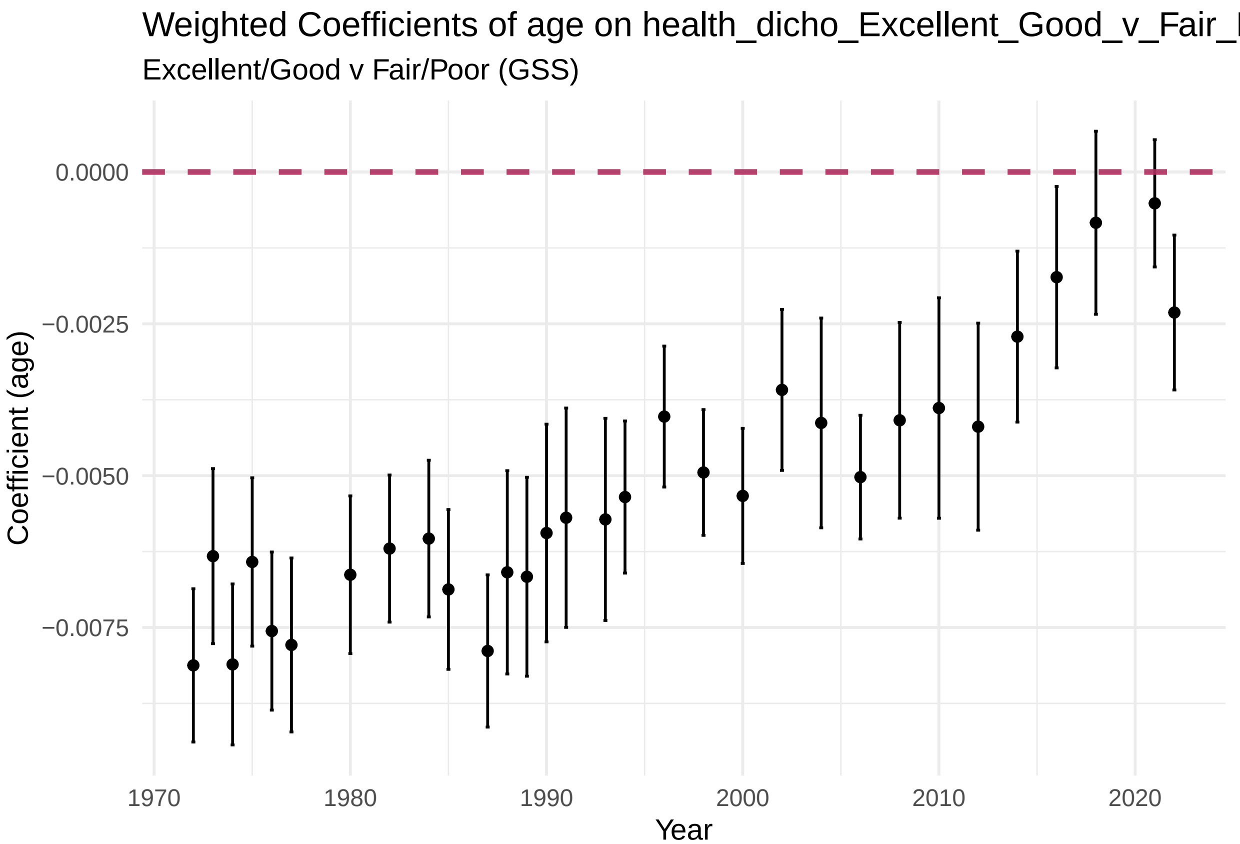
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1972	age	-0.0081234	0.0006432	-12.6286559	0.0000000	-0.0093851	-0.0068617
1973	age	-0.0063250	0.0007348	-8.6075414	0.0000000	-0.0077664	-0.0048836
1974	age	-0.0081079	0.0006755	-12.0019248	0.0000000	-0.0094330	-0.0067828
1975	age	-0.0064208	0.0007066	-9.0873413	0.0000000	-0.0078068	-0.0050348
1976	age	-0.0075585	0.0006635	-11.3919870	0.0000000	-0.0088600	-0.0062570
1977	age	-0.0077872	0.0007299	-10.6685081	0.0000000	-0.0092190	-0.0063554
1980	age	-0.0066315	0.0006620	-10.0170698	0.0000000	-0.0079301	-0.0053328
1982	age	-0.0062002	0.0006173	-10.0438452	0.0000000	-0.0074109	-0.0049895
1984	age	-0.0060359	0.0006573	-9.1821576	0.0000000	-0.0073253	-0.0047464
1985	age	-0.0068734	0.0006707	-10.2479467	0.0000000	-0.0081890	-0.0055578
1987	age	-0.0078861	0.0006390	-12.3417539	0.0000000	-0.0091394	-0.0066329

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1988	age	-0.0065912	0.0008527	-7.7299382	0.0000000	-0.0082645	-0.0049178
1989	age	-0.0066643	0.0008344	-7.9870391	0.0000000	-0.0083016	-0.0050270
1990	age	-0.0059443	0.0009130	-6.5108831	0.0000000	-0.0077361	-0.0041525
1991	age	-0.0056925	0.0009200	-6.1877437	0.0000000	-0.0074978	-0.0038872
1993	age	-0.0057204	0.0008482	-6.7439673	0.0000000	-0.0073848	-0.0040560
1994	age	-0.0053519	0.0006383	-8.3849188	0.0000000	-0.0066036	-0.0041001
1996	age	-0.0040271	0.0005916	-6.8072215	0.0000000	-0.0051872	-0.0028671
1998	age	-0.0049476	0.0005281	-9.3687692	0.0000000	-0.0059831	-0.0039121
2000	age	-0.0053332	0.0005672	-9.4031432	0.0000000	-0.0064455	-0.0042210
2002	age	-0.0035877	0.0006760	-5.3071435	0.0000001	-0.0049135	-0.0022619
2004	age	-0.0041318	0.0008801	-4.6947245	0.0000029	-0.0058583	-0.0024053
2006	age	-0.0050242	0.0005193	-9.6749292	0.0000000	-0.0060423	-0.0040060
2008	age	-0.0040884	0.0008214	-4.9772345	0.0000007	-0.0056998	-0.0024770
2010	age	-0.0038865	0.0009251	-4.2012706	0.0000284	-0.0057013	-0.0020717
2012	age	-0.0041935	0.0008689	-4.8262142	0.0000016	-0.0058982	-0.0024889
2014	age	-0.0027108	0.0007174	-3.7786330	0.0001631	-0.0041179	-0.0013037
2016	age	-0.0017321	0.0007614	-2.2748107	0.0230298	-0.0032254	-0.0002388
2018	age	-0.0008360	0.0007684	-1.0880361	0.2767467	-0.0023432	0.0006711
2021	age	-0.0005161	0.0005343	-0.9660199	0.3340974	-0.0015636	0.0005314
2022	age	-0.0023139	0.0006501	-3.5593263	0.0003770	-0.0035886	-0.0010393

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

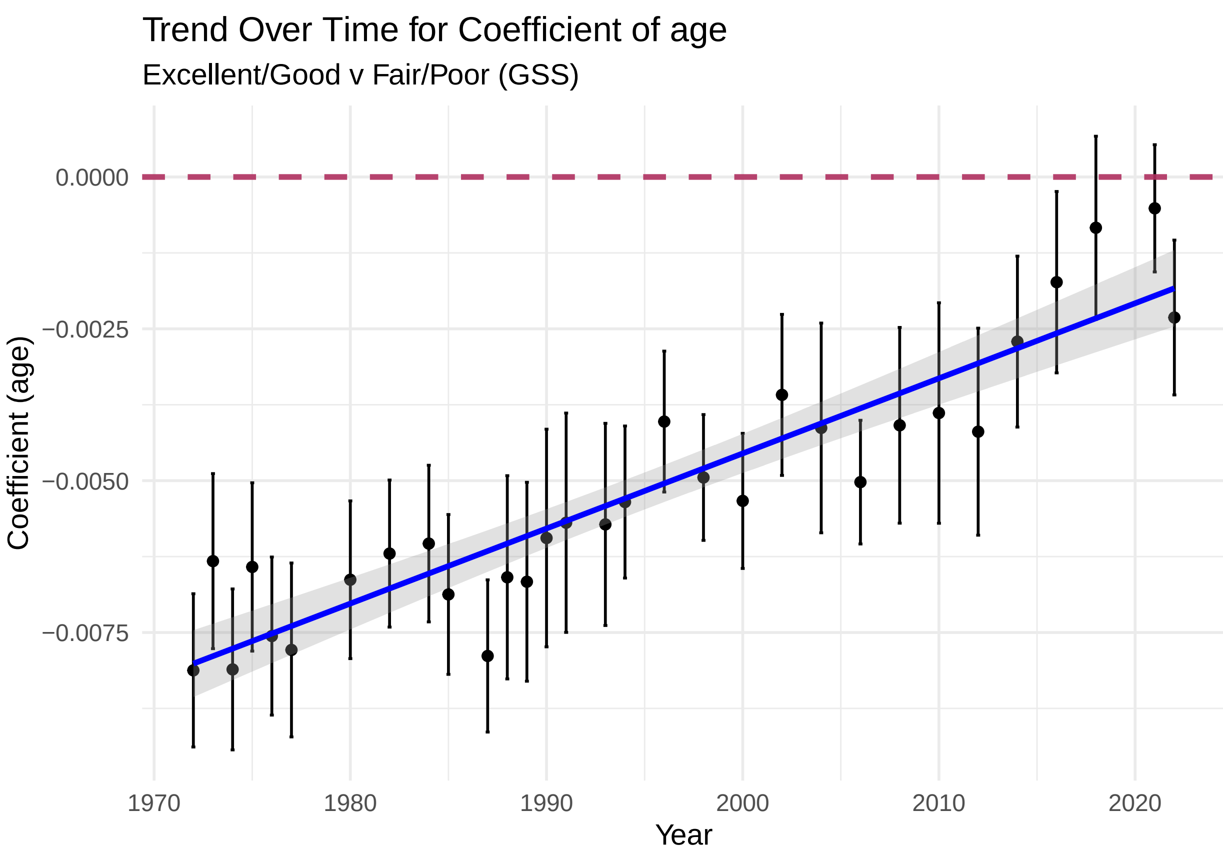
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0017277 -0.0005039 -0.0001108  0.0005348  0.0015640
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.518e-01  1.969e-02  -12.78 1.92e-13 ***
## year         1.236e-04  9.873e-06   12.52 3.21e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0008291 on 29 degrees of freedom
## Multiple R-squared:  0.8439, Adjusted R-squared:  0.8385
## F-statistic: 156.8 on 1 and 29 DF,  p-value: 3.208e-13
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent/Good v Fair/Poor (GSS)"))
```




```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent/Good v Fair/Poor (GSS)"))
```

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




```r
p_coeff_gss_eg_vs_fp <- result_list$coef_trend_plot + labs(subtitle = "Excellent/Good vs Fair/Poor (GSS
```

```r
# health_dicho_Excellent_Good_Fair_v_Poor

# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = gss_svy,    # your srvyr design object
  outcome     = "health_dicho_Excellent_Good_Fair_v_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent/Good/Fair v Poor (GSS)")
```

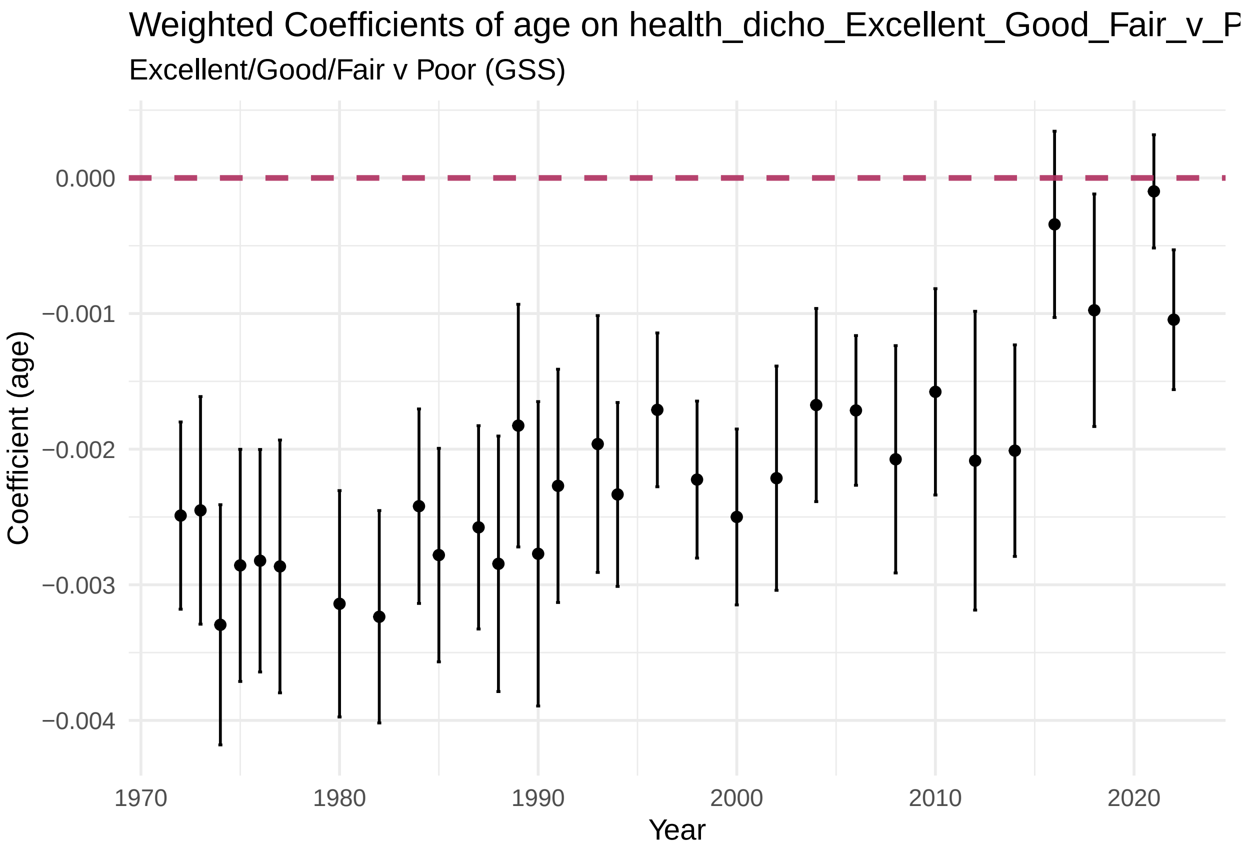
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1972	age	-0.0024897	0.0003520	-7.0735014	0.0000000	-0.0031801	-0.0017993
1973	age	-0.0024512	0.0004278	-5.7300324	0.0000000	-0.0032904	-0.0016121
1974	age	-0.0032951	0.0004515	-7.2986887	0.0000000	-0.0041807	-0.0024095
1975	age	-0.0028570	0.0004363	-6.5480968	0.0000000	-0.0037129	-0.0020011
1976	age	-0.0028226	0.0004182	-6.7496429	0.0000000	-0.0036429	-0.0020023
1977	age	-0.0028646	0.0004751	-6.0295161	0.0000000	-0.0037965	-0.0019327
1980	age	-0.0031401	0.0004254	-7.3816285	0.0000000	-0.0039746	-0.0023057
1982	age	-0.0032358	0.0003993	-8.1043475	0.0000000	-0.0040189	-0.0024527
1984	age	-0.0024204	0.0003655	-6.6219240	0.0000000	-0.0031374	-0.0017034
1985	age	-0.0027809	0.0004013	-6.9295418	0.0000000	-0.0035681	-0.0019937
1987	age	-0.0025764	0.0003822	-6.7419470	0.0000000	-0.0033259	-0.0018269

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
1988	age	-0.0028452	0.0004801	-5.9266558	0.0000000	-0.0037873	-0.0019031
1989	age	-0.0018263	0.0004559	-4.0057685	0.0000663	-0.0027209	-0.0009317
1990	age	-0.0027714	0.0005718	-4.8467231	0.0000015	-0.0038937	-0.0016492
1991	age	-0.0022706	0.0004382	-5.1809948	0.0000003	-0.0031306	-0.0014105
1993	age	-0.0019620	0.0004826	-4.0658464	0.0000514	-0.0029088	-0.0010151
1994	age	-0.0023339	0.0003458	-6.7502027	0.0000000	-0.0030120	-0.0016558
1996	age	-0.0017100	0.0002892	-5.9133480	0.0000000	-0.0022771	-0.0011430
1998	age	-0.0022243	0.0002951	-7.5370767	0.0000000	-0.0028029	-0.0016456
2000	age	-0.0024999	0.0003306	-7.5609521	0.0000000	-0.0031482	-0.0018515
2002	age	-0.0022139	0.0004216	-5.2514697	0.0000002	-0.0030407	-0.0013871
2004	age	-0.0016745	0.0003630	-4.6132340	0.0000043	-0.0023866	-0.0009625
2006	age	-0.0017142	0.0002815	-6.0899286	0.0000000	-0.0022661	-0.0011623
2008	age	-0.0020746	0.0004272	-4.8557521	0.0000013	-0.0029127	-0.0012365
2010	age	-0.0015772	0.0003878	-4.0671954	0.0000505	-0.0023380	-0.0008164
2012	age	-0.0020851	0.0005615	-3.7133614	0.0002132	-0.0031866	-0.0009835
2014	age	-0.0020109	0.0003976	-5.0572452	0.0000005	-0.0027908	-0.0012310
2016	age	-0.0003422	0.0003502	-0.9773832	0.3285056	-0.0010290	0.0003445
2018	age	-0.0009756	0.0004371	-2.2321156	0.0257484	-0.0018330	-0.0001183
2021	age	-0.0000991	0.0002129	-0.4656059	0.6415250	-0.0005166	0.0003183
2022	age	-0.0010454	0.0002629	-3.9767151	0.0000714	-0.0015608	-0.0005300

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

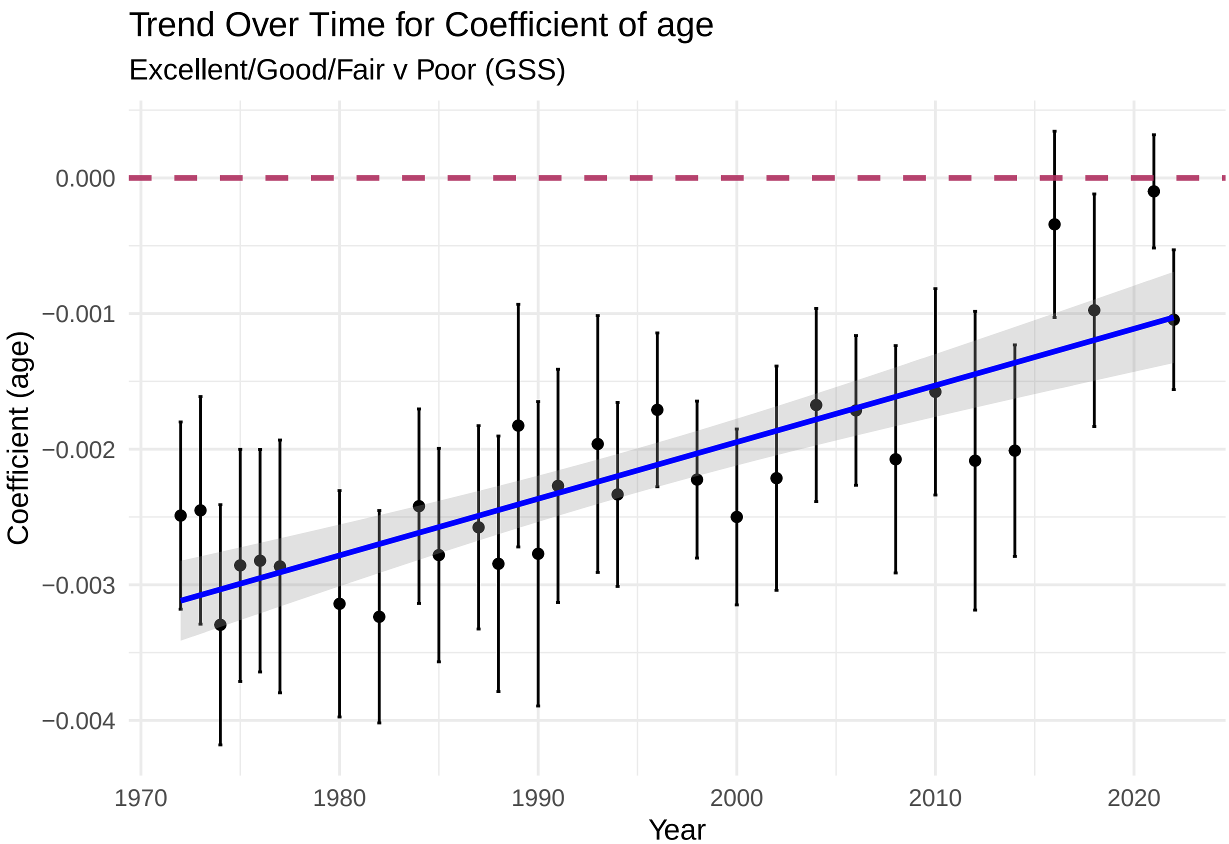
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.481e-04 -3.532e-04 -1.715e-05  2.079e-04  9.712e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.551e-02  1.053e-02  -8.118 5.95e-09 ***
## year         4.178e-05  5.281e-06   7.913 1.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0004434 on 29 degrees of freedom
## Multiple R-squared:  0.6834, Adjusted R-squared:  0.6725
## F-statistic: 62.61 on 1 and 29 DF,  p-value: 1e-08
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent/Good/Fair v Poor (GSS)"))
```




```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent/Good/Fair v Poor (GSS)"))
```

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




```
p_coeff_gss_egf_vs_p <- result_list$coef_trend_plot + labs(subtitle = "Excellent/Good/Fair vs Poor (GSS)
```

NHANES

Load and Wrangle Data

```
# Load and wrangle nhanes data

data_nh <- read_csv(here("code_examples/kamaryn_nhanes/nhanes_1999-2018_2023-11-29.csv"))
```

```
## New names:
## Rows: 96241 Columns: 226
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): sex dbl (223): ...1, X, BaseID, visit, id, age_visit, race, whas_1_2,
## age_exam, ... lgl (2): date_visit, date_last
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
dim(data_nh)
```

```
## [1] 96241   226
```

```
sum(is.na(data_nh$health)) # note there are 38579 NA's for health out of 96241 subj
```

```
## [1] 38578
```

```r
year_to_wave <- read_csv(here("big_data/NHANES/nhanes_4/nh4_year_to_wave.csv")) %>%
  mutate(release_nb = wave)
```

```
## Rows: 10 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): year, wave
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
data_nhanes <- data_nh %>%
  filter(visit == 1) %>%
  filter(age_visit >= 18) %>%
  filter(!(is.na(health))) %>% # remove subjects without our primary variable of interest
 # left_join(year_to_wave, by = "release_nb") %>% # year of survey (***note each survey is two years, t
  mutate(srh = 6 - health) %>%  # recode SRH to be more intuitive (Excellent = 5 to Poor = 1)
  mutate(age = age_visit) # more intuitive naming for APC analyses

# variables of interest:
# self-rated health: "health" originally, reverse recoded to "srh"
# age at survey: "age_visit"
# age at follow-up (censored/deceased): "age_last"
# vital status at follow-up (alive/decesased): "deceased"
# year/wave: "release_nb"
# SEQN: "BaseID"
# "ddem_wghts"  Full Sample 2 Year MEC Exam Weight  WTMEC2YR
# "dem_wghts_4yr"   Full Sample 4 Year MEC Exam Weight  WTMEC4YR


# Adding gloria's for weight variable: SDDSRVYR and other clock and APC things if needed

data_nhanes_gloria <- read.csv(here("big_data/NHANES/Gloria_preprocessed/Data-3/Core_Dataset_Aim2.csv"))

data_nhanes <- data_nhanes %>%
  mutate(SEQN = as.character(BaseID)) %>%
  left_join(data_nhanes_gloria %>%
              rename(race_decoded = race) %>%
              mutate(age = as.numeric(age)),
            by = c("SEQN", "age")) %>%
  mutate(age = as.numeric(age))

# add more

data_nhanes <- data_nhanes %>%
  mutate(age_group = as.factor(
                            cut(
                            age,
                            breaks = c(17, 29, 39, 49, 59, 69, Inf),
                            labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                            levels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                            right = TRUE
                          )),
         health_cat = factor(srh,
```

```
                        levels = 1:5,
                        labels = c("Poor", "Fair", "Good", "Very Good", "Excellent")))
```

```
# more new columns to enable APC
data_nhanes <- data_nhanes %>%
  mutate(period_cut_6 = as.factor(cut(data_nhanes$year, 6)),
    period_cut_10 = as.factor(cut(data_nhanes$year, 10)),
    period_cut_12 = as.factor(cut(data_nhanes$year, 12)),
    period_groups = as.factor(cut(data_nhanes$year, 12)),
    period_10yr = as.factor(
                            cut(
                            year,
                            breaks = c(1973, 1982, 1990, 1998, 2006, 2014, Inf),
                            labels = c("1974-1982", "1982-1990", "1990-1998",
                                       "1998-2006", "2006-2014", "2014-2022"),
                            right = TRUE
                            )
                          ),
    period_decade = as.factor(
                            cut(
                            year,
                            breaks = c(1973, 1979, 1989, 1999, 2009, 2019, Inf),
                            labels = c("1974-1979", "1980-1989", "1990-1999",
                                       "2000-2009", "2010-2019", "2020-2024"),
                            right = TRUE
                            )
                          ),
    age_group = as.factor(
                            cut(
                            age,
                            breaks = c(17, 29, 39, 49, 59, 69, Inf),
                            labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                            right = TRUE
                          )),
    age_groups = as.factor(
                            cut(
                            age,
                            breaks = c(17, 29, 39, 49, 59, 69, Inf),
                            labels = c("18-29", "30-39", "40-49", "50-59", "60-69", "70+"),
                            right = TRUE
                          )),
    age_group_small = as.factor(
                            cut(
                              age,
                              breaks = c(seq(15, 75, by = 5), Inf),  # Define breaks up to 75 and inclu
                              labels = c("16-20", "21-25", "26-30", "31-35", "36-40", "41-45", "46-50",
                              right = FALSE  # Makes intervals left-closed, i.e., [x, y)
                            )
                          )
                          ,
    generation = factor(
      case_when(
        cohort >= 1901 & cohort <= 1927 ~ "Greatest (1901-1927)",
        cohort >= 1928 & cohort <= 1945 ~ "Silent (1928-1945)",
```

```
      cohort >= 1946 & cohort <= 1964 ~ "Boomers (1946-1964)",
      cohort >= 1965 & cohort <= 1980 ~ "Gen X (1965-1980)",
      cohort >= 1981 & cohort <= 1996 ~ "Millennials (1981-1996)",
      cohort >= 1997 & cohort <= 2012 ~ "Gen Z (1997-2012)",
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest (1901-1927)",
      "Silent (1928-1945)",
      "Boomers (1946-1964)",
      "Gen X (1965-1980)",
      "Millennials (1981-1996)",
      "Gen Z (1997-2012)"#,
   #   "Other"
    )
  ),
  generation_two_sections = factor(
    case_when(
      generation == "Greatest (1901-1927)" & cohort <= 1914 ~ "Greatest Early (1901-1914)",
      generation == "Greatest (1901-1927)" & cohort > 1914 ~ "Greatest Late (1915-1927)",
      generation == "Silent (1928-1945)" & cohort <= 1936 ~ "Silent Early (1928-1936)",
      generation == "Silent (1928-1945)" & cohort > 1936 ~ "Silent Late (1937-1945)",
      generation == "Boomers (1946-1964)" & cohort <= 1955 ~ "Boomers Early (1946-1955)",
      generation == "Boomers (1946-1964)" & cohort > 1955 ~ "Boomers Late (1956-1964)",
      generation == "Gen X (1965-1980)" & cohort <= 1972 ~ "Gen X Early (1965-1972)",
      generation == "Gen X (1965-1980)" & cohort > 1972 ~ "Gen X Late (1973-1980)",
      generation == "Millennials (1981-1996)" & cohort <= 1988 ~ "Millennials Early (1981-1988)",
      generation == "Millennials (1981-1996)" & cohort > 1988 ~ "Millennials Late (1989-1996)",
      generation == "Gen Z (1997-2012)" & cohort <= 2004 ~ "Gen Z Early (1997-2004)",
      generation == "Gen Z (1997-2012)" & cohort > 2004 ~ "Gen Z Late (2005-2012)",
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest Early (1901-1914)", "Greatest Late (1915-1927)",
      "Silent Early (1928-1936)", "Silent Late (1937-1945)",
      "Boomers Early (1946-1955)", "Boomers Late (1956-1964)",
      "Gen X Early (1965-1972)", "Gen X Late (1973-1980)",
      "Millennials Early (1981-1988)", "Millennials Late (1989-1996)",
      "Gen Z Early (1997-2004)", "Gen Z Late (2005-2012)"#,
   #   "Other"
    )
  ),
  generation_three_sections = factor(
    case_when(
      generation == "Greatest (1901-1927)" & cohort <= 1910 ~ "Greatest Early (1901-1910)",
      generation == "Greatest (1901-1927)" & cohort > 1910 & cohort <= 1918 ~ "Greatest Mid (1911-1918
      generation == "Greatest (1901-1927)" & cohort > 1918 ~ "Greatest Late (1919-1927)",
      generation == "Silent (1928-1945)" & cohort <= 1934 ~ "Silent Early (1928-1934)",
      generation == "Silent (1928-1945)" & cohort > 1934 & cohort <= 1940 ~ "Silent Mid (1935-1940)",
      generation == "Silent (1928-1945)" & cohort > 1940 ~ "Silent Late (1941-1945)",
      generation == "Boomers (1946-1964)" & cohort <= 1951 ~ "Boomers Early (1946-1951)",
      generation == "Boomers (1946-1964)" & cohort > 1951 & cohort <= 1958 ~ "Boomers Mid (1952-1958)
      generation == "Boomers (1946-1964)" & cohort > 1958 ~ "Boomers Late (1959-1964)",
```

```
      generation == "Gen X (1965-1980)" & cohort <= 1970 ~ "Gen X Early (1965-1970)",
      generation == "Gen X (1965-1980)" & cohort > 1970 & cohort <= 1976 ~ "Gen X Mid (1971-1976)",
      generation == "Gen X (1965-1980)" & cohort > 1976 ~ "Gen X Late (1977-1980)",
      generation == "Millennials (1981-1996)" & cohort <= 1986 ~ "Millennials Early (1981-1986)",
      generation == "Millennials (1981-1996)" & cohort > 1986 & cohort <= 1992 ~ "Millennials Mid (19
      generation == "Millennials (1981-1996)" & cohort > 1992 ~ "Millennials Late / Gen Z (1993-2004)
  #    generation == "Gen Z (1997-2012)" & cohort <= 2002 ~ "Gen Z Early (1997-2002)",
  #    generation == "Gen Z (1997-2012)" & cohort > 2002 & cohort <= 2008 ~ "Gen Z Mid (2003-2008)",
  #    generation == "Gen Z (1997-2012)" & cohort > 2008 ~ "Gen Z Late (2009-2012)",
      TRUE ~ "Other"
    ),
    levels = c(
      "Greatest Early (1901-1910)", "Greatest Mid (1911-1918)", "Greatest Late (1919-1927)",
      "Silent Early (1928-1934)", "Silent Mid (1935-1940)", "Silent Late (1941-1945)",
      "Boomers Early (1946-1951)", "Boomers Mid (1952-1958)", "Boomers Late (1959-1964)",
      "Gen X Early (1965-1970)", "Gen X Mid (1971-1976)", "Gen X Late (1977-1980)",
      "Millennials Early (1981-1986)", "Millennials Mid (1987-1992)",
      "Millennials Late / Gen Z (1993-2004)"
      #"Millennials Late (1993-1996)",
    #  "Gen Z Early (1997-2002)", "Gen Z Mid (2003-2008)", "Gen Z Late (2009-2012)" #,
    #  "Other"
    )
  ))
```

Dichotomize

```
# Define the full set of categories
all_levels <- c("Excellent", "Very Good", "Good", "Fair", "Poor")

# Use the combinatorial 'combn' function to get all subsets
all_subsets <- map(1:length(all_levels), ~combn(all_levels, m = .x, simplify = FALSE)) %>%
  # Flatten the list of lists
  flatten()

# # Filter out the empty set (already not returned by combn) and the full set
# # Keep everything else: these are the potential "Group 1"s
# valid_subsets <- all_subsets %>%
#   discard(~length(.x) == 5)  # remove subset with all 4 categories

# delete the weird ones

valid_subsets <- all_subsets[c(1:6, 16, 26)]

# Create dichotomized variables

# A helper function to turn a subset (S) into a meaningful column name
make_dicho_name <- function(S) {
  # Example name: "Ex_Go" vs "Fa_Po" for c("Excellent","Good"), etc.
  # We'll join group names with underscores and separate them with "v".
  group1_name <- str_c(S, collapse = "_")
  # We find the complement
  group2 <- setdiff(all_levels, S)
```

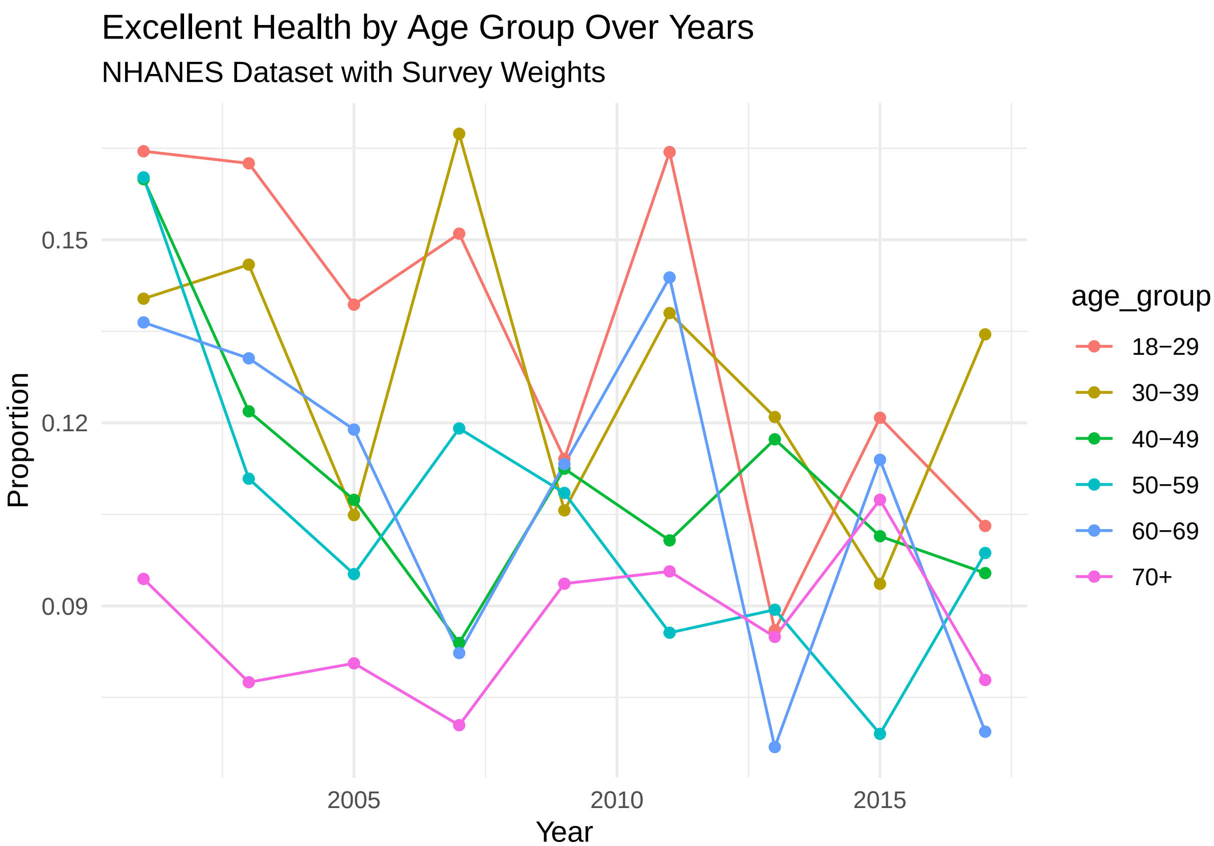
```r
  group2_name <- str_c(group2, collapse = "_")
  # Combine them
  out_name <- str_c("health_dicho_", group1_name, "_v_", group2_name)
  # For cleanliness, ensure no weird characters/spaces:
  out_name <- str_replace_all(out_name, "\\s+", "")
  out_name
}

data_nhanes_dichotomized <- data_nhanes

for (subset_i in valid_subsets) {

  new_col_name <- make_dicho_name(subset_i)

  # Dichotomize: if health_cat is in 'subset_i', call it "Group1", else "Group2"
  data_nhanes_dichotomized <- data_nhanes_dichotomized %>%
    mutate(
      !!sym(new_col_name) := if_else(health_cat %in% subset_i,
                                      1, 0) #"Group1",
                                      #"Group2")
    )
}

data_nhanes <- data_nhanes_dichotomized

dichotomized_var <- colnames(data_nhanes[(length(data_nhanes) - length(valid_subsets)):(length(data_nhan

important_dichotomized_var_5levels <- c("health_dicho_Excellent_v_VeryGood_Good_Fair_Poor",
            "health_dicho_Excellent_VeryGood_v_Good_Fair_Poor",
            "health_dicho_Excellent_VeryGood_Good_v_Fair_Poor",
            "health_dicho_Excellent_VeryGood_Good_Fair_v_Poor"
                                       )
```

Survey object

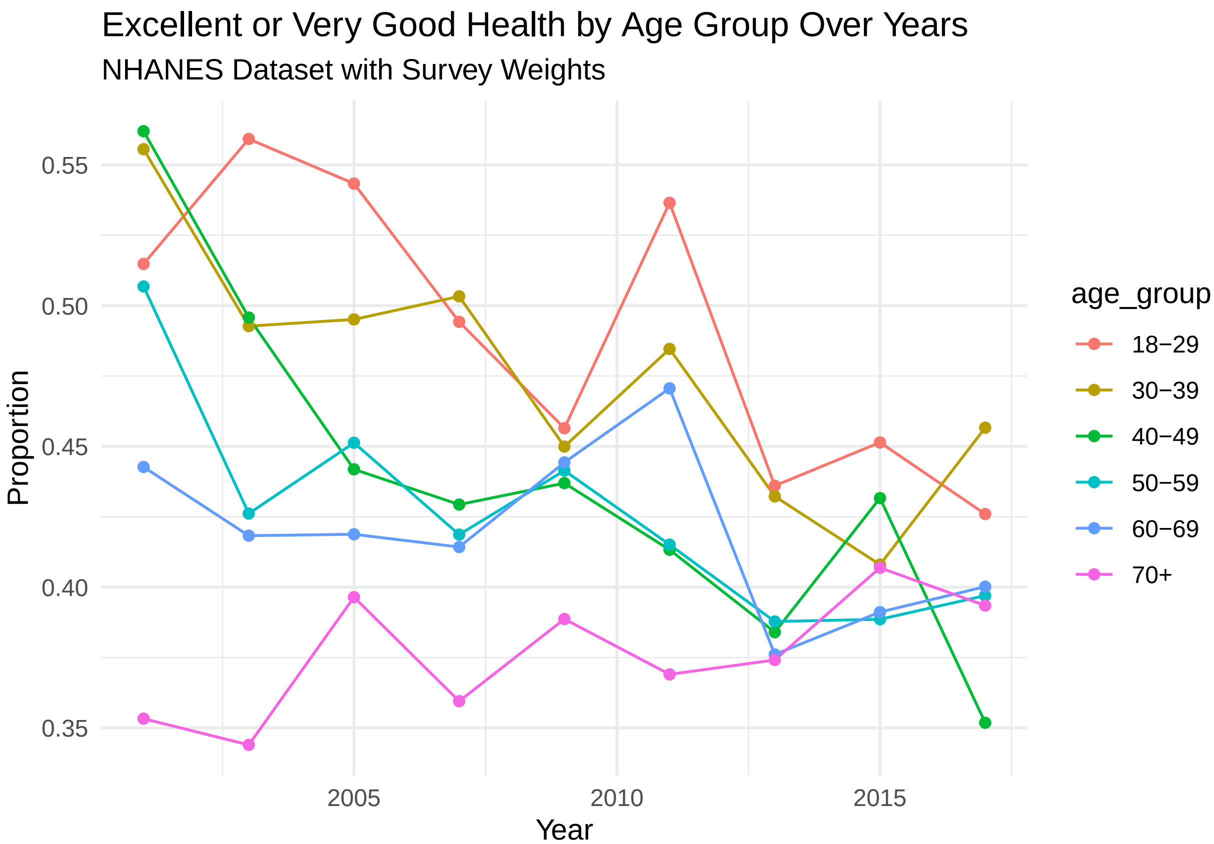
```r
library(survey)
library(srvyr)


svy_nhanes <- data_nhanes %>%
  filter(!(is.na(SDMVPSU))) %>%
  as_survey_design(
  #  ids = 1,
    ids = SDMVPSU,           # PSU identifiers (use 1 if not available)
  #  weights = WTINT2YR,  # original -- interview weights -- larger sample size but fewer covariates
    weights = WTMEC2YR, # Gloria's -- enable more covariates
    strata = SDMVSTRA,
    nest = TRUE
    )
```

First figure

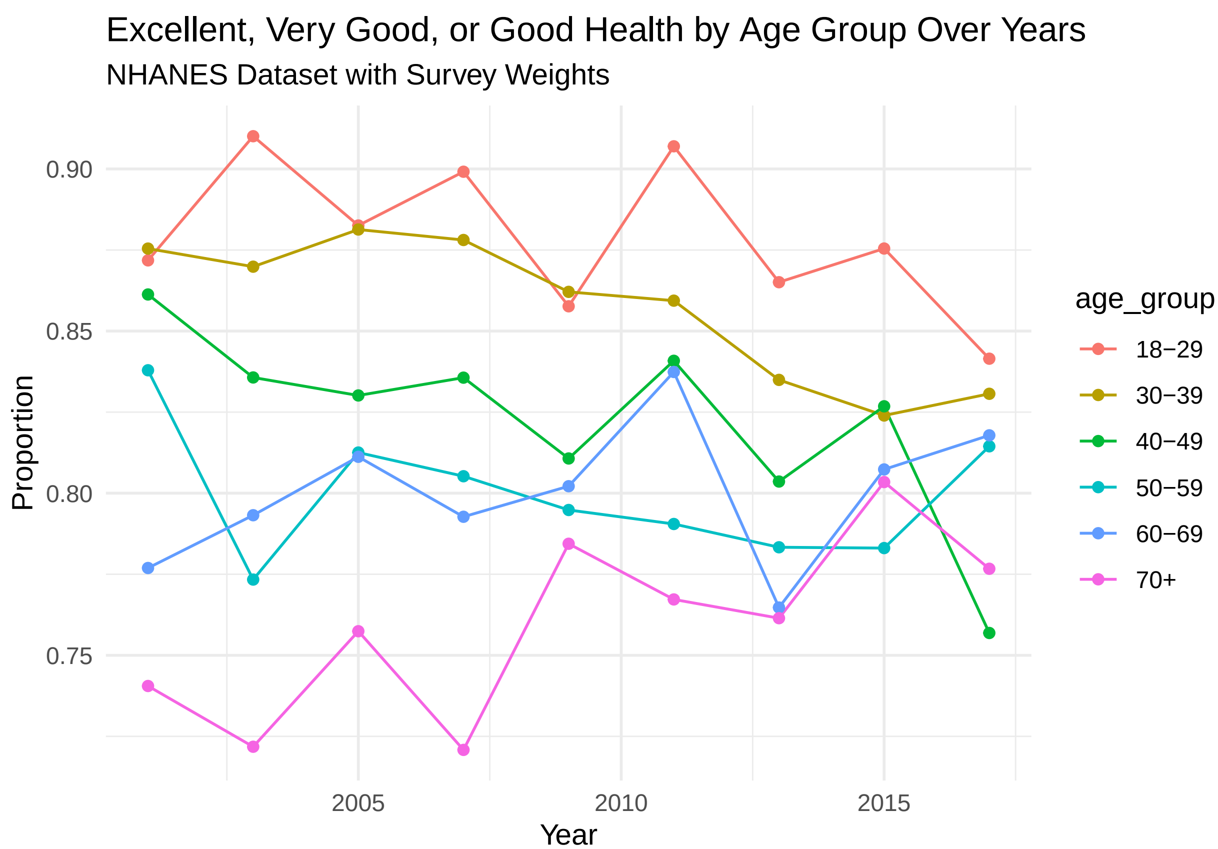
```
#"health_dicho_Excellent_v_VeryGood_Good_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_v_Good_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_Good_v_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_Good_Fair_v_Poor"

p_prop_nhanes_e_vs_vgfp <- plot_weighted_proportion(
  data = svy_nhanes,
  outcome = health_dicho_Excellent_v_VeryGood_Good_Fair_Poor,
  group_var = age_group,
  year_var = year,
  title = "Excellent Health by Age Group Over Years",
  subtitle = "NHANES Dataset with Survey Weights"
)
p_prop_nhanes_e_vs_vgfp
```



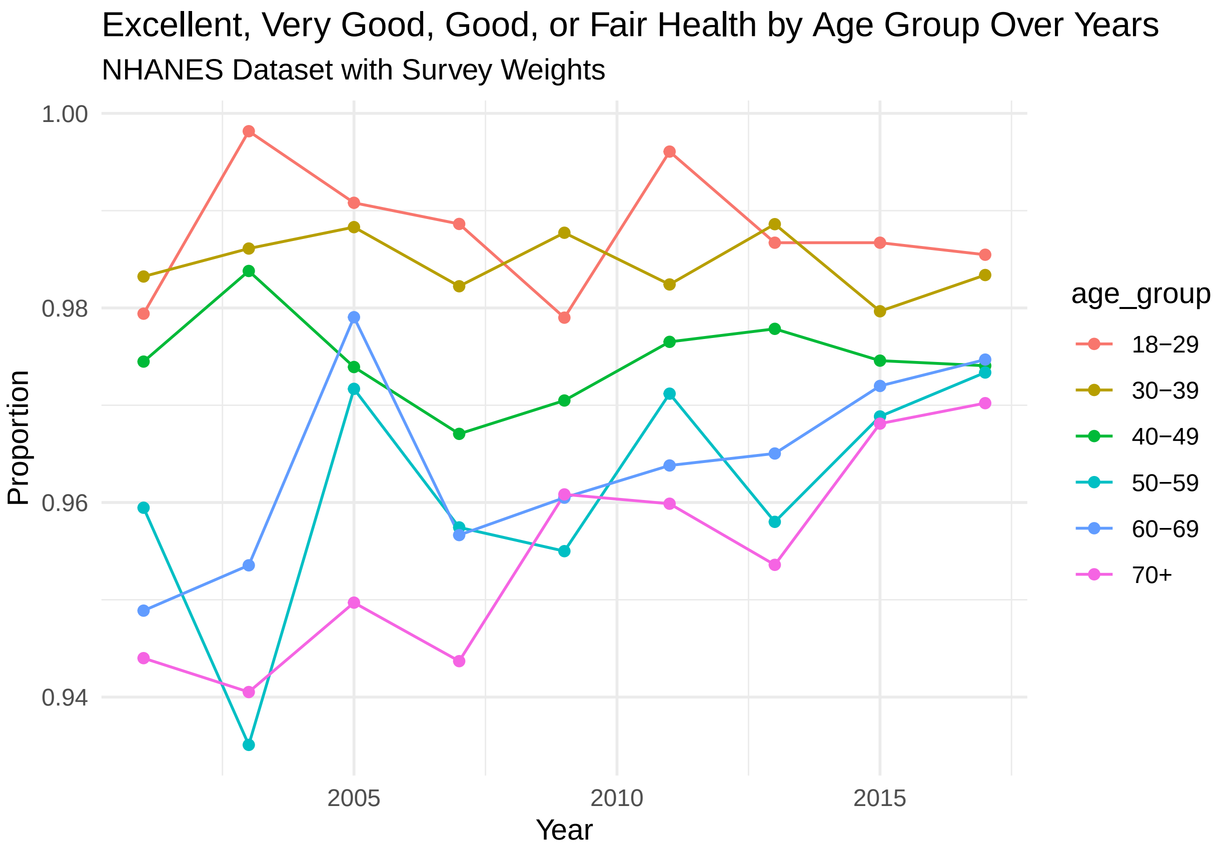

```
p_prop_nhanes_ev_vs_gfp <- plot_weighted_proportion(
  data = svy_nhanes,
  outcome = health_dicho_Excellent_VeryGood_v_Good_Fair_Poor,
  group_var = age_group,
  year_var = year,
  title = "Excellent or Very Good Health by Age Group Over Years",
  subtitle = "NHANES Dataset with Survey Weights"
)
p_prop_nhanes_ev_vs_gfp
```




```r
p_prop_nhanes_evg_vs_fp <- plot_weighted_proportion(
  data = svy_nhanes,
  outcome = health_dicho_Excellent_VeryGood_Good_v_Fair_Poor,
  group_var = age_group,
  year_var = year,
  title = "Excellent, Very Good, or Good Health by Age Group Over Years",
  subtitle = "NHANES Dataset with Survey Weights"
)
p_prop_nhanes_evg_vs_fp
```




```r
p_prop_nhanes_evgf_vs_p <- plot_weighted_proportion(
  data = svy_nhanes,
  outcome = health_dicho_Excellent_VeryGood_Good_Fair_v_Poor,
  group_var = age_group,
  year_var = year,
  title = "Excellent, Very Good, Good, or Fair Health by Age Group Over Years",
  subtitle = "NHANES Dataset with Survey Weights")

  p_prop_nhanes_evgf_vs_p
```



Age Coeff

```r
#"health_dicho_Excellent_v_VeryGood_Good_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_v_Good_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_Good_v_Fair_Poor",
#             "health_dicho_Excellent_VeryGood_Good_Fair_v_Poor"

# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = svy_nhanes,    # your srvyr design object
  outcome     = "health_dicho_Excellent_v_VeryGood_Good_Fair_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent v VG/Good/Fair/Poor (NHANES)")
```

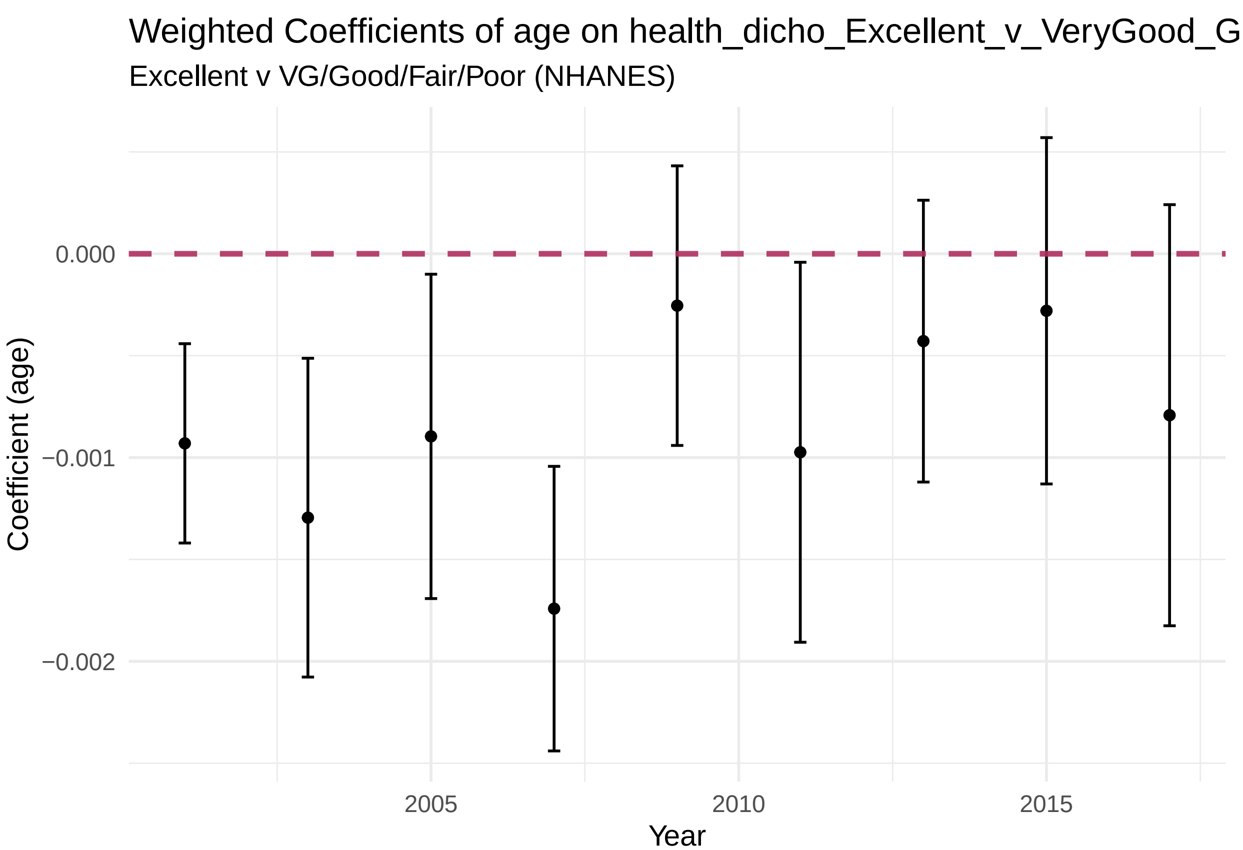
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2001	age	-0.0009303	0.0002281	-4.0791971	0.0011270	-0.0014195	-0.0004412
2003	age	-0.0012950	0.0003647	-3.5507405	0.0031962	-0.0020772	-0.0005128
2005	age	-0.0008961	0.0003711	-2.4146394	0.0300149	-0.0016920	-0.0001001
2007	age	-0.0017415	0.0003276	-5.3152524	0.0000865	-0.0024398	-0.0010431
2009	age	-0.0002545	0.0003219	-0.7905600	0.4415213	-0.0009405	0.0004316
2011	age	-0.0009739	0.0004399	-2.2141195	0.0416880	-0.0019064	-0.0000414

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2013	age	-0.0004286	0.0003224	-1.3291870	0.2050417	-0.0011201	0.0002630
2015	age	-0.0002797	0.0003963	-0.7057642	0.4919125	-0.0011295	0.0005702
2017	age	-0.0007921	0.0004818	-1.6441224	0.1224105	-0.0018255	0.0002412

```
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

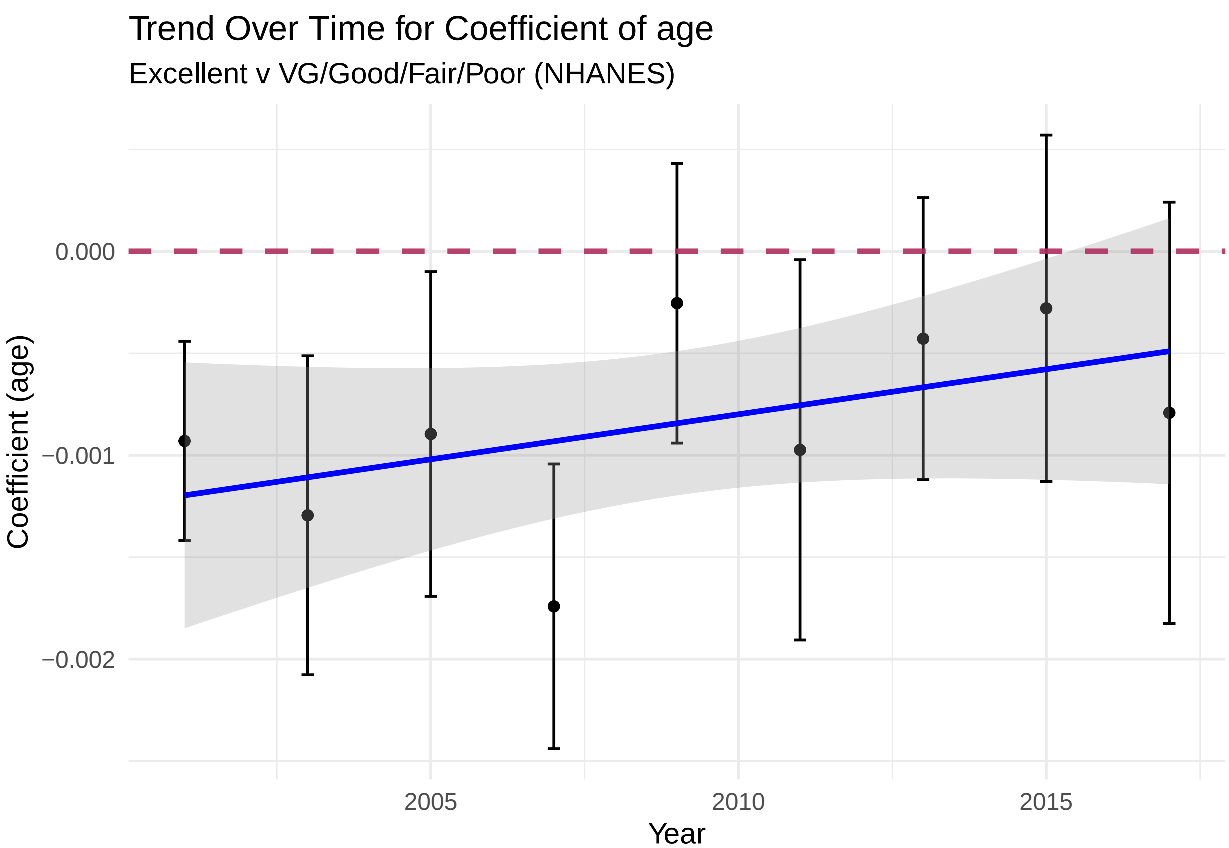
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0008096 -0.0002188  0.0001242  0.0002666  0.0005891
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.960e-02  5.818e-02  -1.540    0.167
## year         4.418e-05  2.896e-05   1.526    0.171
##
## Residual standard error: 0.0004486 on 7 degrees of freedom
## Multiple R-squared:  0.2495, Adjusted R-squared:  0.1423
## F-statistic: 2.327 on 1 and 7 DF,  p-value: 0.171
```

```
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent v VG/Good/Fair/Poor (NHANES)"))
```



```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent v VG/Good/Fair/Poor (NHANES)"))
```

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




```r
p_coeff_nhanes_e_vs_vgfp <- result_list$coef_trend_plot + labs(subtitle = "Excellent v VG/Good/Fair/Poo


##########


# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = svy_nhanes,    # your srvyr design object
  outcome     = "health_dicho_Excellent_VeryGood_v_Good_Fair_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent/VG v Good/Fair/Poor (NHANES)")
```

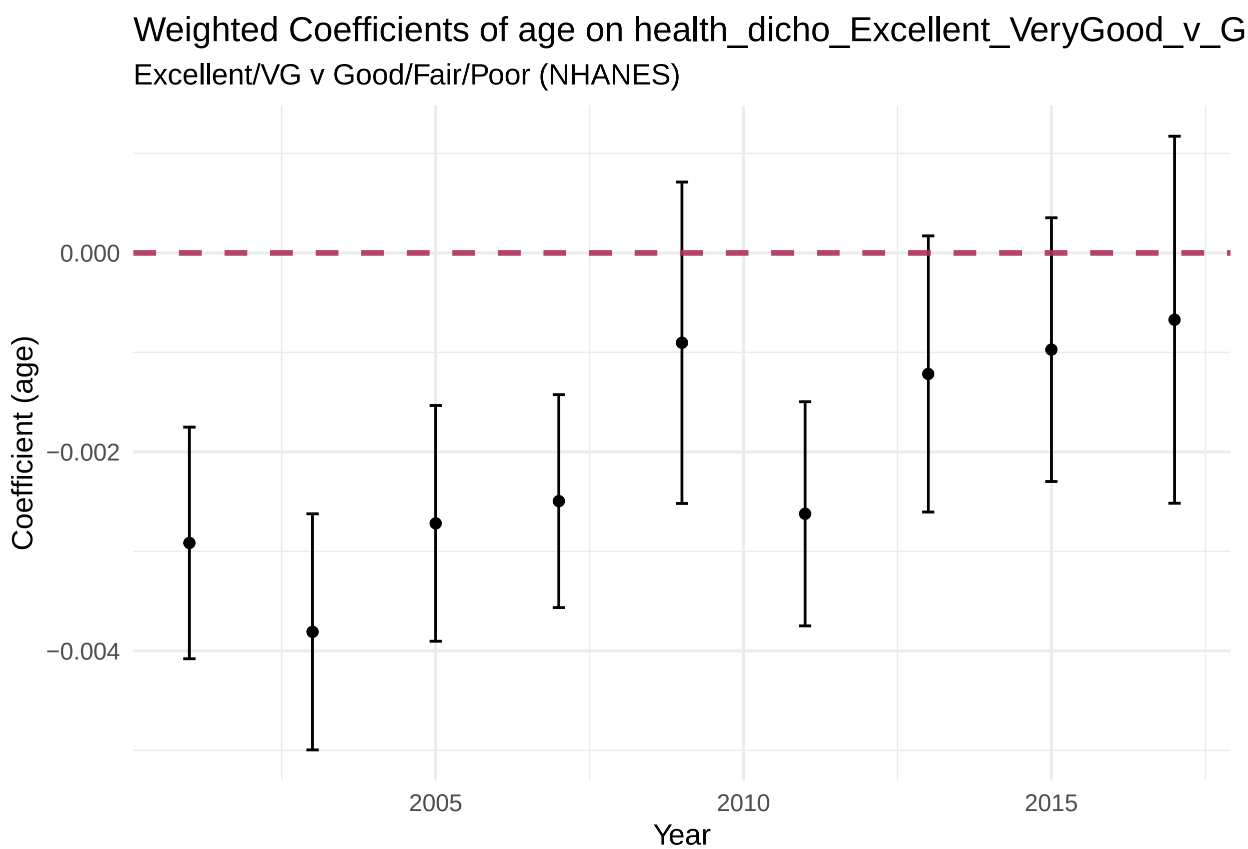
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2001	age	-0.0029153	0.0005427	-5.3715136	0.0000985	-0.0040793	-0.0017512
2003	age	-0.0038089	0.0005533	-6.8838369	0.0000075	-0.0049956	-0.0026222
2005	age	-0.0027186	0.0005525	-4.9203536	0.0002256	-0.0039036	-0.0015335
2007	age	-0.0024949	0.0005021	-4.9691170	0.0001681	-0.0035651	-0.0014248

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2009	age	-0.0009030	0.0007579	-1.1914134	0.2520023	-0.0025184	0.0007125
2011	age	-0.0026222	0.0005314	-4.9340482	0.0001495	-0.0037488	-0.0014956
2013	age	-0.0012164	0.0006472	-1.8796073	0.0811363	-0.0026044	0.0001716
2015	age	-0.0009722	0.0006181	-1.5729121	0.1380603	-0.0022979	0.0003535
2017	age	-0.0006715	0.0008601	-0.7807662	0.4479414	-0.0025162	0.0011731

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

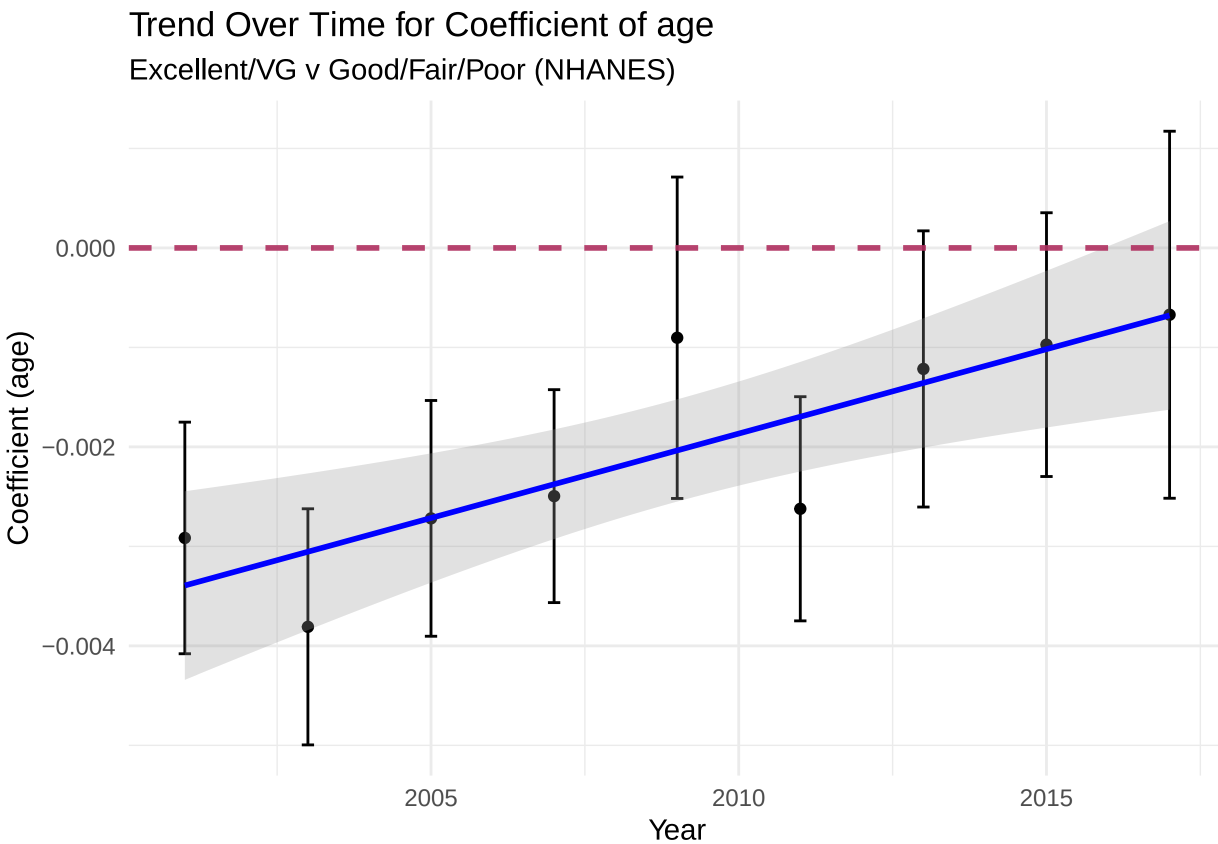
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0009257 -0.0001197  0.0000069  0.0001407  0.0011329
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.429e-01  8.454e-02  -4.056  0.00483 **
## year         1.697e-04  4.208e-05   4.032  0.00498 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0006519 on 7 degrees of freedom
## Multiple R-squared:  0.699,  Adjusted R-squared:  0.656
## F-statistic: 16.26 on 1 and 7 DF,  p-value: 0.004982
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent/VG v Good/Fair/Poor (NHANES)"))
```




```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent/VG v Good/Fair/Poor (NHANES)"))
```

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



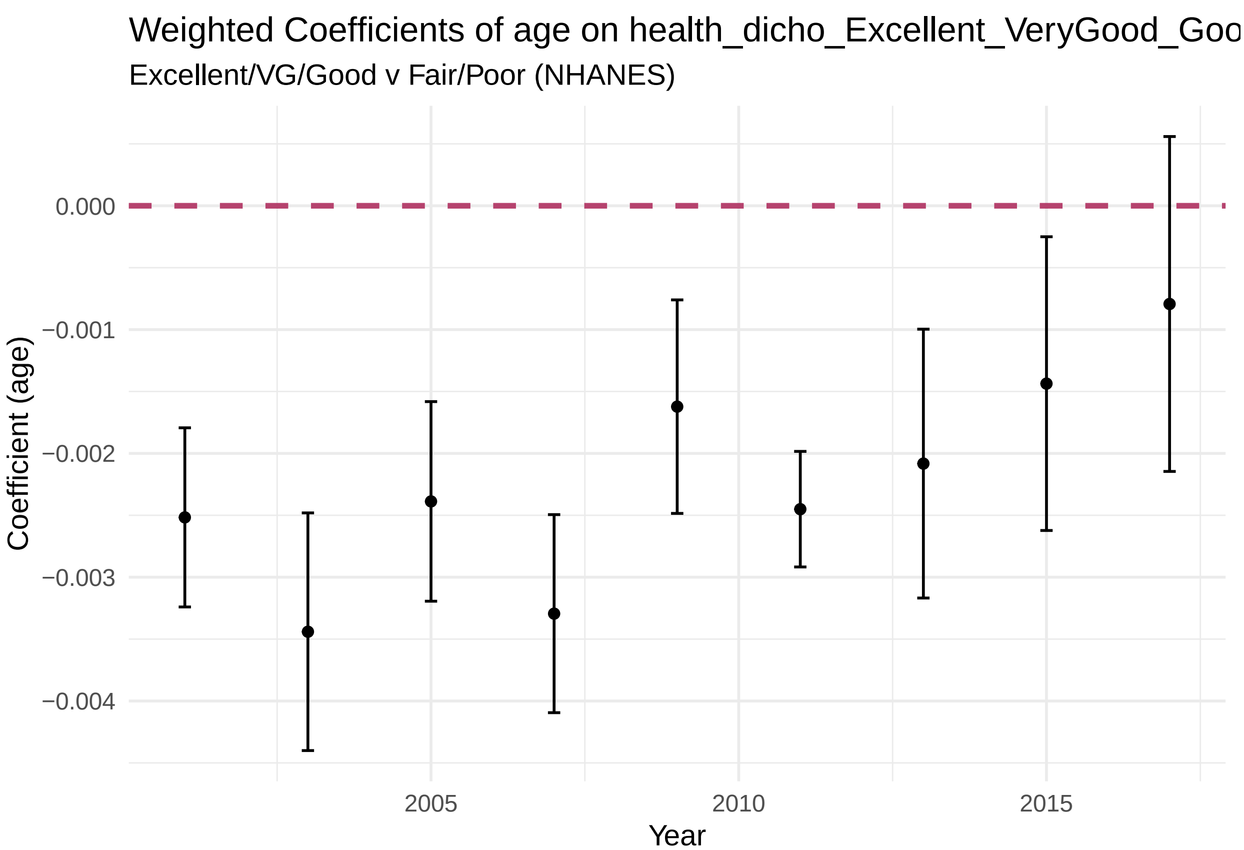

```r
p_coeff_nhanes_ev_vs_gfp <- result_list$coef_trend_plot + labs(subtitle = "Excellent/VG v Good/Fair/Poo

#########


# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = svy_nhanes,    # your srvyr design object
  outcome     = "health_dicho_Excellent_VeryGood_Good_v_Fair_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent/VG/Good v Fair/Poor (NHANES)")
```

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2001	age	-0.0025169	0.0003374	-7.460385	0.0000031	-0.0032405	-0.0017933
2003	age	-0.0034406	0.0004475	-7.688264	0.0000022	-0.0044005	-0.0024808
2005	age	-0.0023879	0.0003758	-6.353560	0.0000179	-0.0031940	-0.0015818
2007	age	-0.0032945	0.0003753	-8.778581	0.0000003	-0.0040944	-0.0024946
2009	age	-0.0016224	0.0004046	-4.010146	0.0011356	-0.0024848	-0.0007601
2011	age	-0.0024506	0.0002201	-11.132665	0.0000000	-0.0029173	-0.0019840
2013	age	-0.0020824	0.0005062	-4.113449	0.0010541	-0.0031682	-0.0009966
2015	age	-0.0014365	0.0005531	-2.597350	0.0210858	-0.0026227	-0.0002503
2017	age	-0.0007932	0.0006306	-1.257796	0.2290411	-0.0021458	0.0005594

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

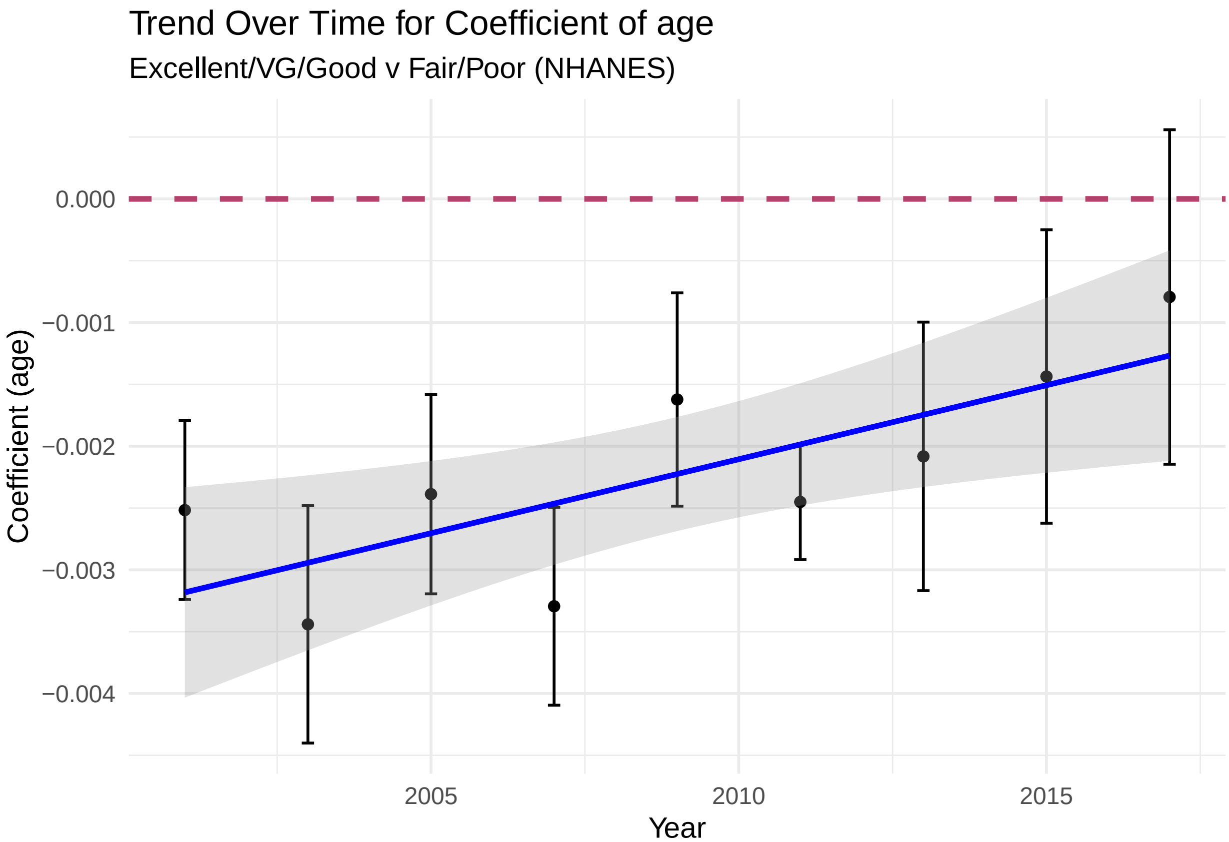
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -8.301e-04 -4.650e-04  7.042e-05  4.743e-04  6.656e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.427e-01  7.596e-02  -3.195   0.0152 *
## year         1.197e-04  3.781e-05   3.165   0.0158 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005857 on 7 degrees of freedom
## Multiple R-squared:  0.5887, Adjusted R-squared:   0.53
## F-statistic: 10.02 on 1 and 7 DF,  p-value: 0.01581
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent/VG/Good v Fair/Poor (NHANES)"))
```




```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent/VG/Good v Fair/Poor (NHANES)"))
```

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




```
p_coeff_nhanes_evg_vs_fp <- result_list$coef_trend_plot + labs(subtitle = "Excellent/VG/Good v Fair/Poo

#######


# 1. Run Weighted Regressions by Year
result_list <- weighted_regressions_by_year(
  survey_data = svy_nhanes,    # your srvyr design object
  outcome     = "health_dicho_Excellent_VeryGood_Good_Fair_v_Poor",  # outcome variable
  predictor   = "age",      # main predictor
  year_var    = "year"      # group by year
)

# 2. Check the extracted coefficient estimates
knitr::kable(result_list$coefficient_df, title = "Excellent/VG/Good/Fair v Poor (NHANES)")
```

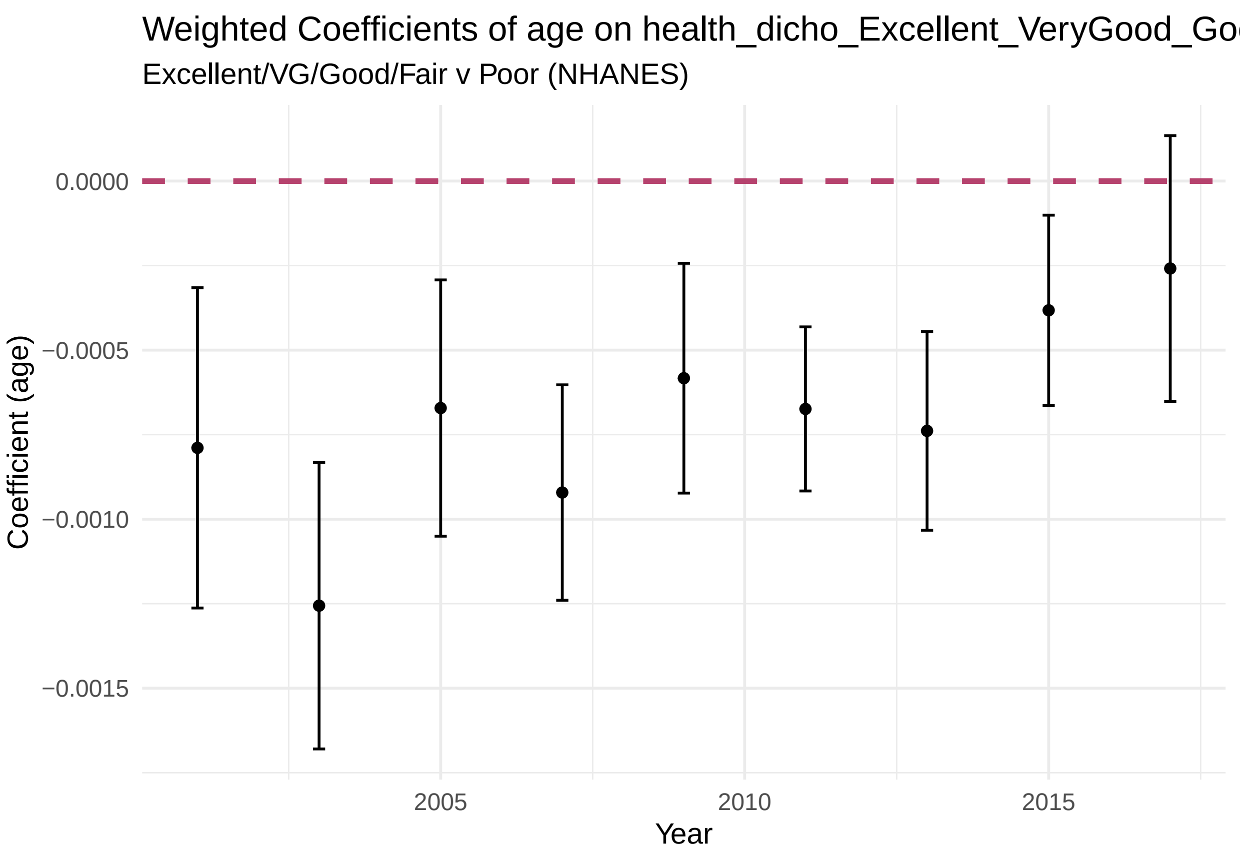
year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2001	age	-0.0007891	0.0002208	-3.572987	0.0030581	-0.0012628	-0.0003154
2003	age	-0.0012558	0.0001976	-6.355449	0.0000178	-0.0016796	-0.0008320
2005	age	-0.0006714	0.0001767	-3.799957	0.0019510	-0.0010503	-0.0002924
2007	age	-0.0009212	0.0001494	-6.164681	0.0000181	-0.0012397	-0.0006027
2009	age	-0.0005831	0.0001594	-3.657243	0.0023349	-0.0009229	-0.0002432
2011	age	-0.0006740	0.0001145	-5.888141	0.0000229	-0.0009167	-0.0004313
2013	age	-0.0007388	0.0001370	-5.392964	0.0000948	-0.0010327	-0.0004450
2015	age	-0.0003823	0.0001312	-2.913986	0.0113246	-0.0006636	-0.0001009

year	term	estimate	std.error	statistic	p.value	conf.low	conf.high
2017	age	-0.0002586	0.0001832	-1.411105	0.1800484	-0.0006516	0.0001344

```r
# 3. View the summary of how the coefficient changes over time
result_list$lm_over_time_summary
```

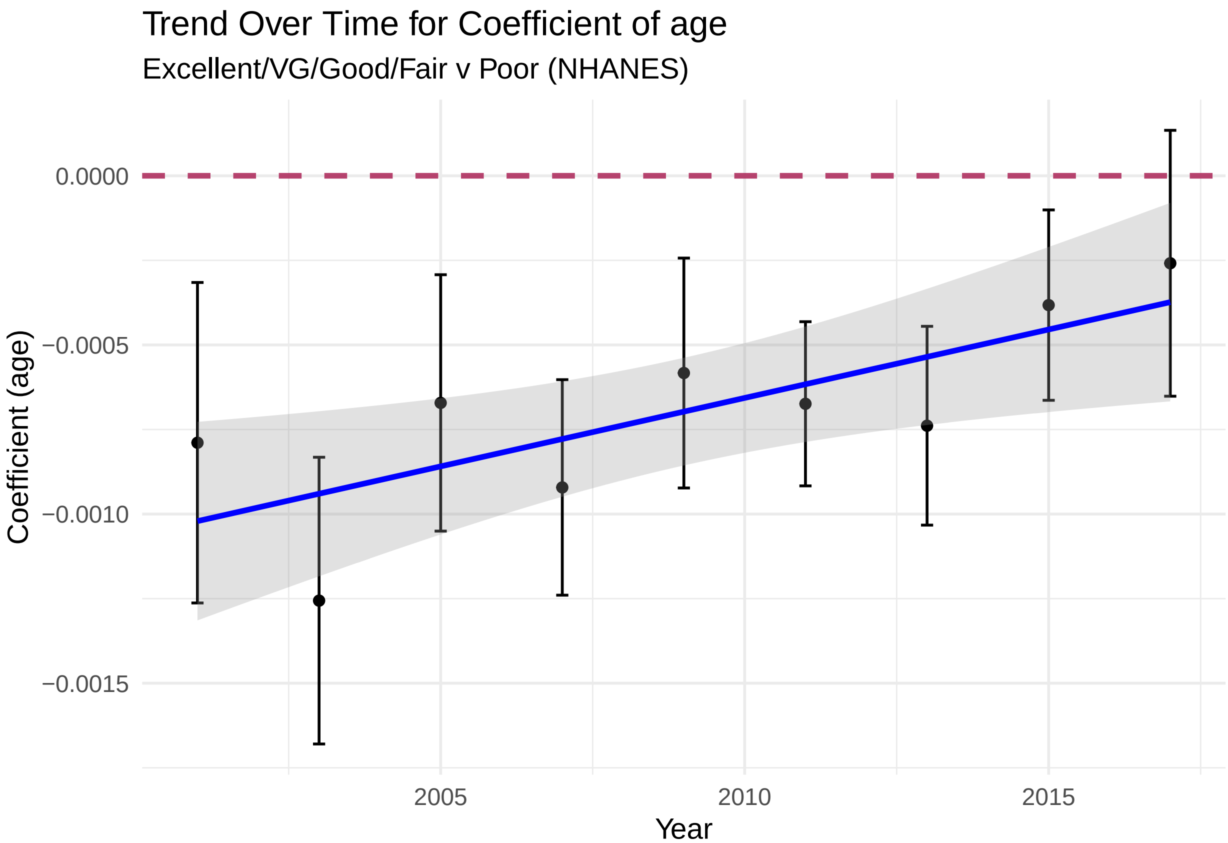
```
##
## Call:
## lm(formula = estimate ~ year, data = coefficient_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.159e-04 -1.432e-04  7.212e-05  1.149e-04  2.317e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.198e-02  2.618e-02  -3.131   0.0166 *
## year         4.046e-05  1.303e-05   3.104   0.0172 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002019 on 7 degrees of freedom
## Multiple R-squared:  0.5792, Adjusted R-squared:  0.5191
## F-statistic: 9.636 on 1 and 7 DF,  p-value: 0.01722
```

```r
# 4. Plot the coefficient estimates across years
print(result_list$coef_plot + labs(subtitle = "Excellent/VG/Good/Fair v Poor (NHANES)"))
```



```r
# 5. Plot the coefficient estimates with a linear trend line
print(result_list$coef_trend_plot + labs(subtitle = "Excellent/VG/Good/Fair v Poor (NHANES)"))
```

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

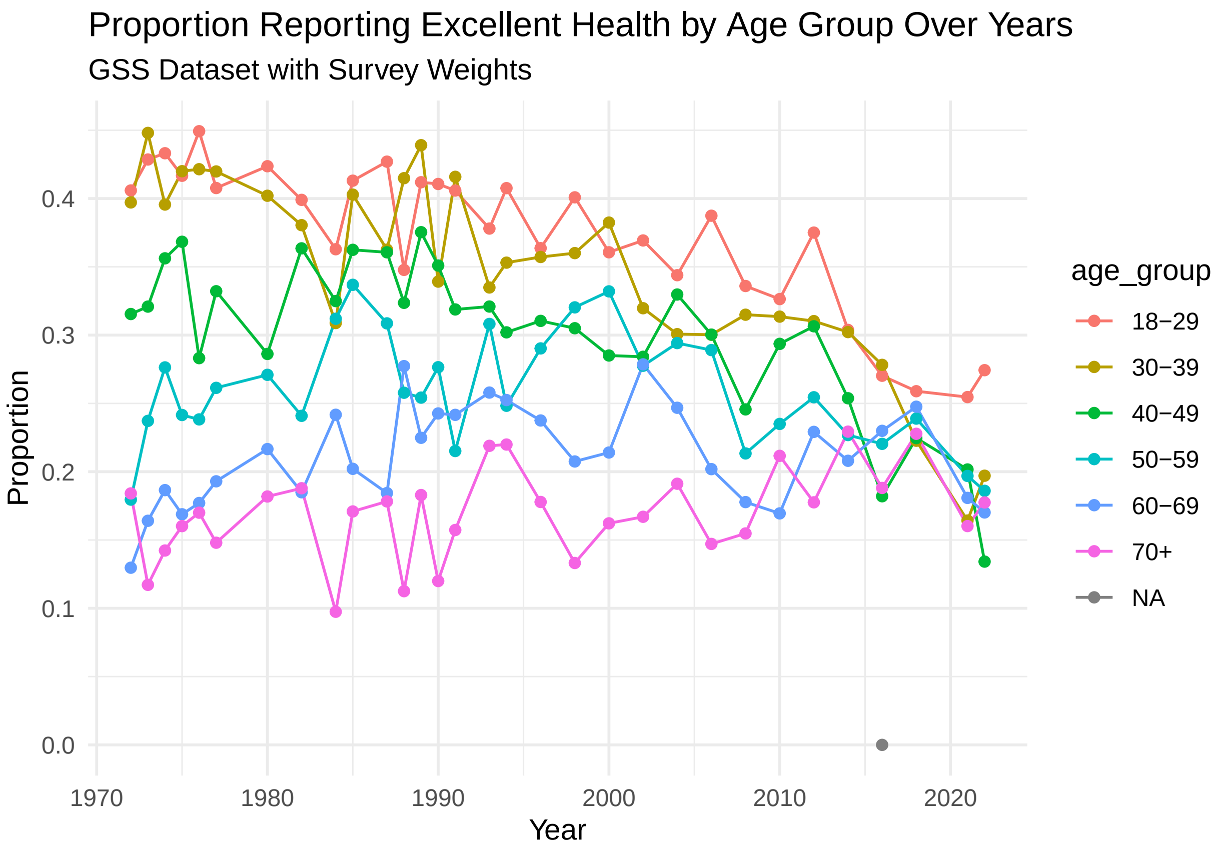



```r
p_coeff_nhanes_evgf_vs_p <- result_list$coef_trend_plot + labs(subtitle = "Excellent/VG/Good/Fair v Poor
```

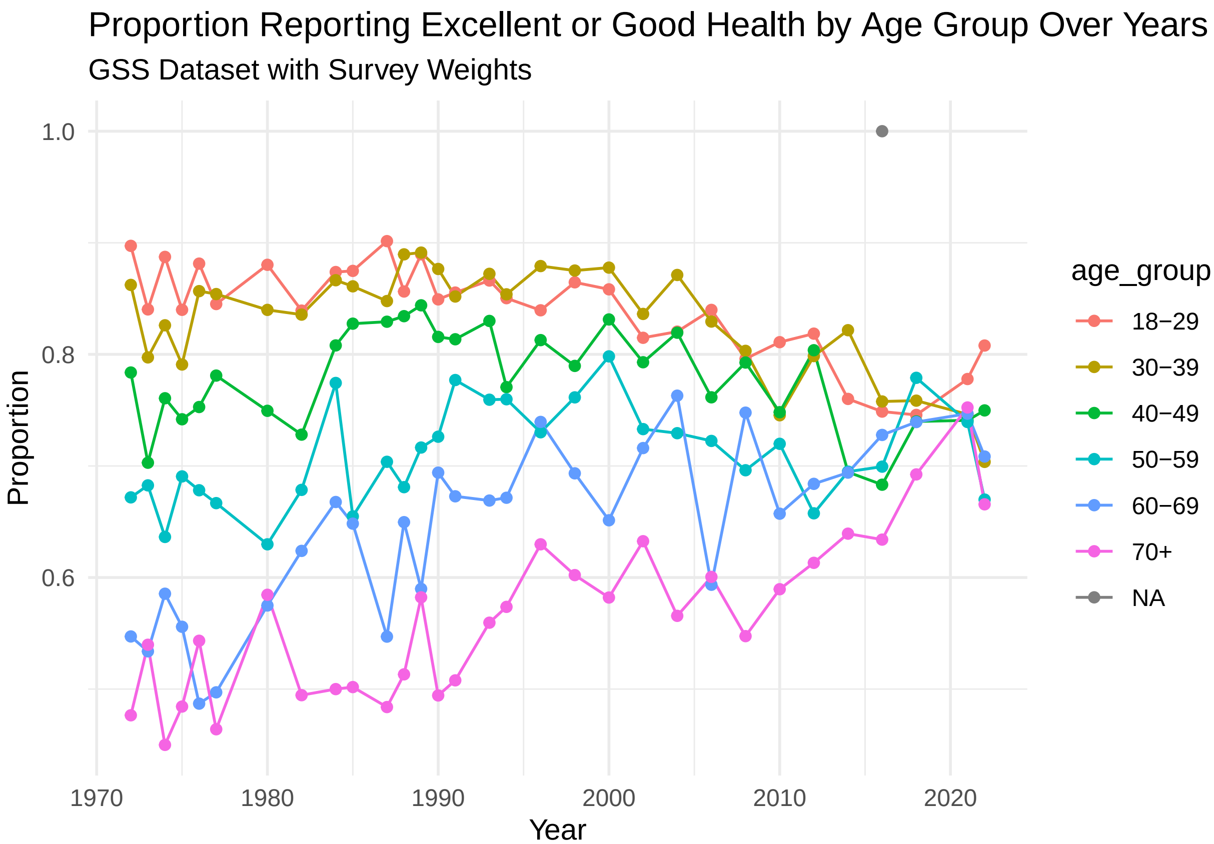
View it all together

GSS

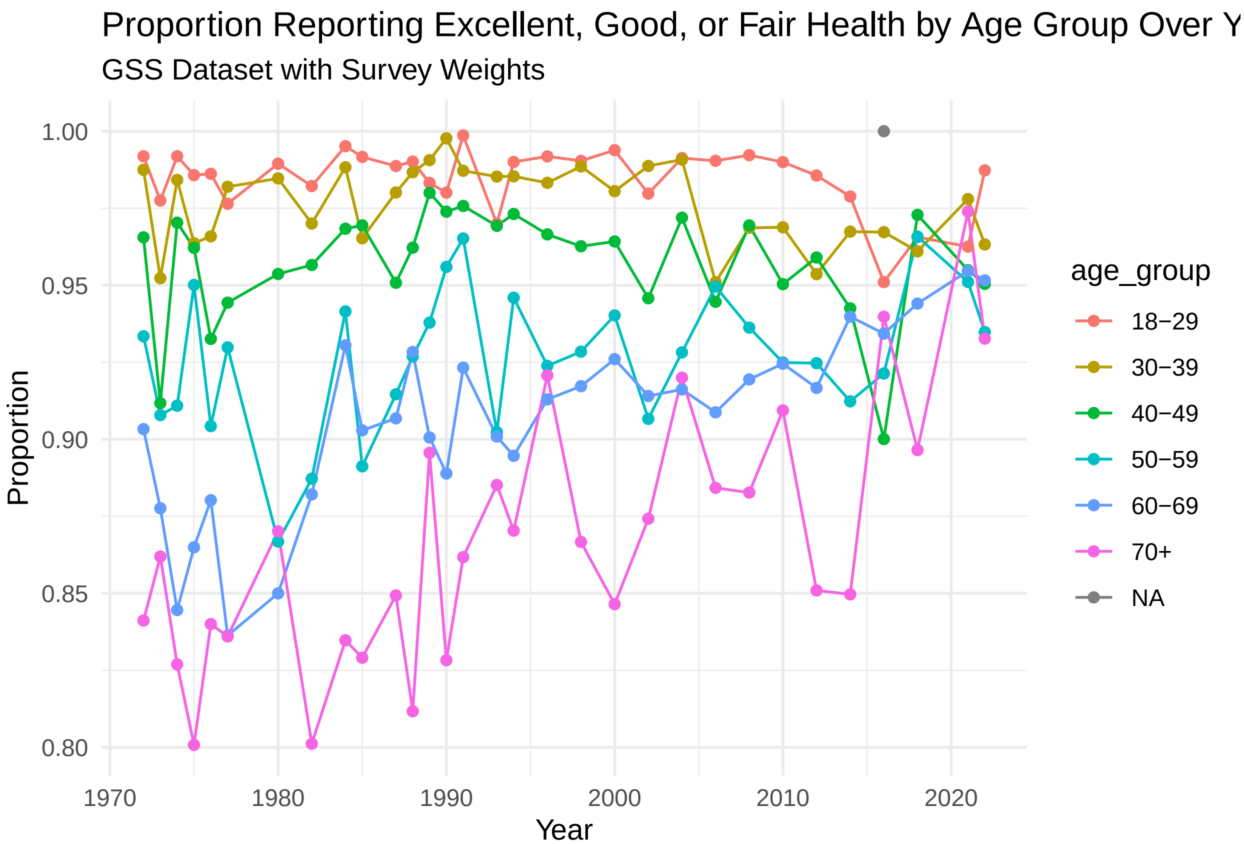
```r
p_prop_gss_e_vs_gfp
```




```
p_prop_gss_eg_vs_fp
```

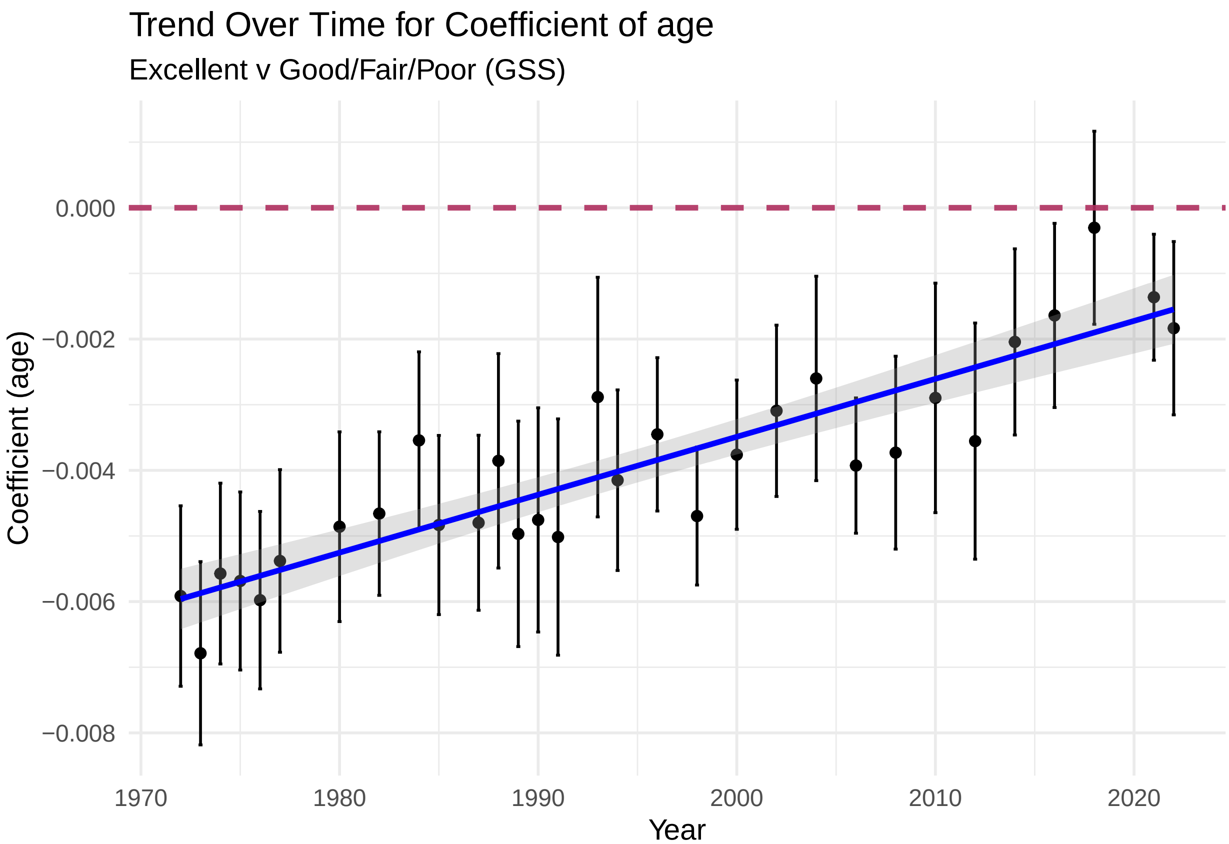



```
p_prop_gss_egf_vs_p
```



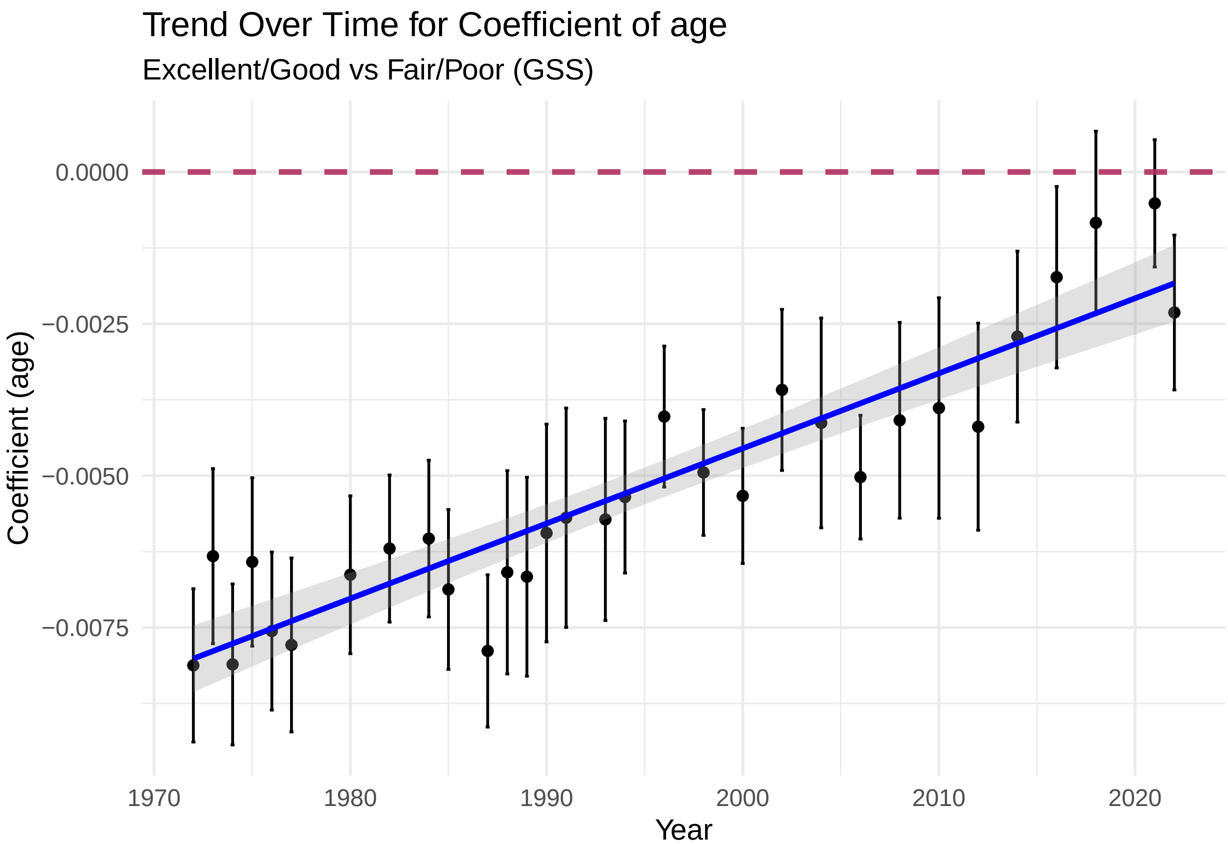

```
p_coeff_gss_e_vs_gfp
```

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



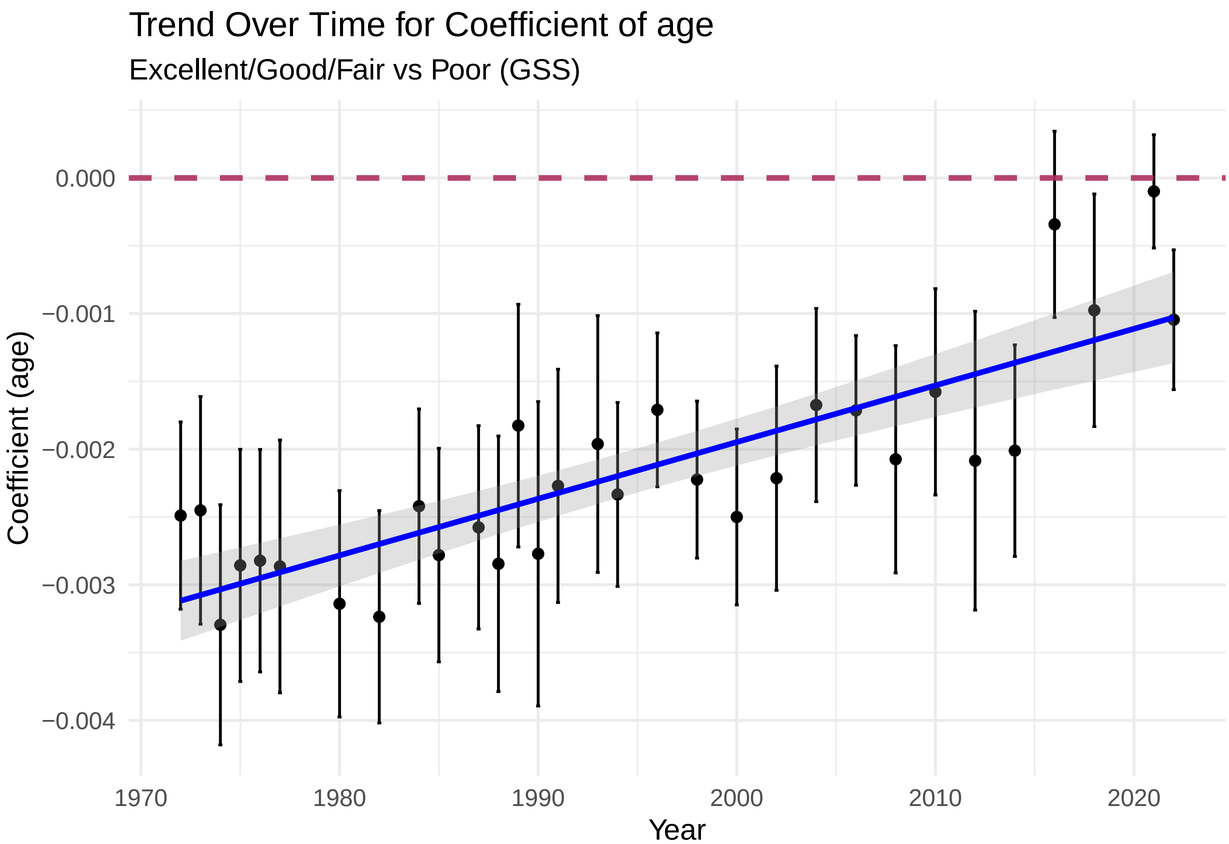

```
p_coeff_gss_eg_vs_fp
```

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



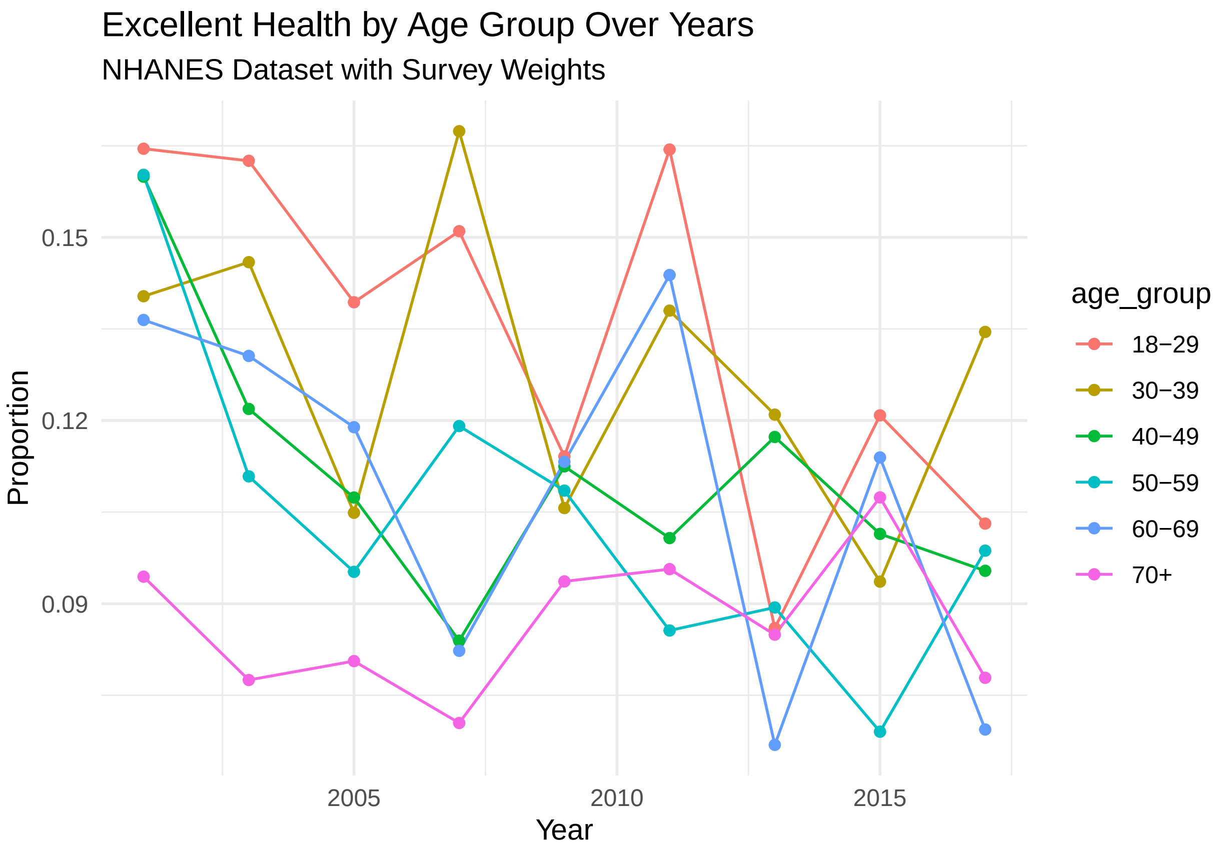

```
p_coeff_gss_egf_vs_p
```

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

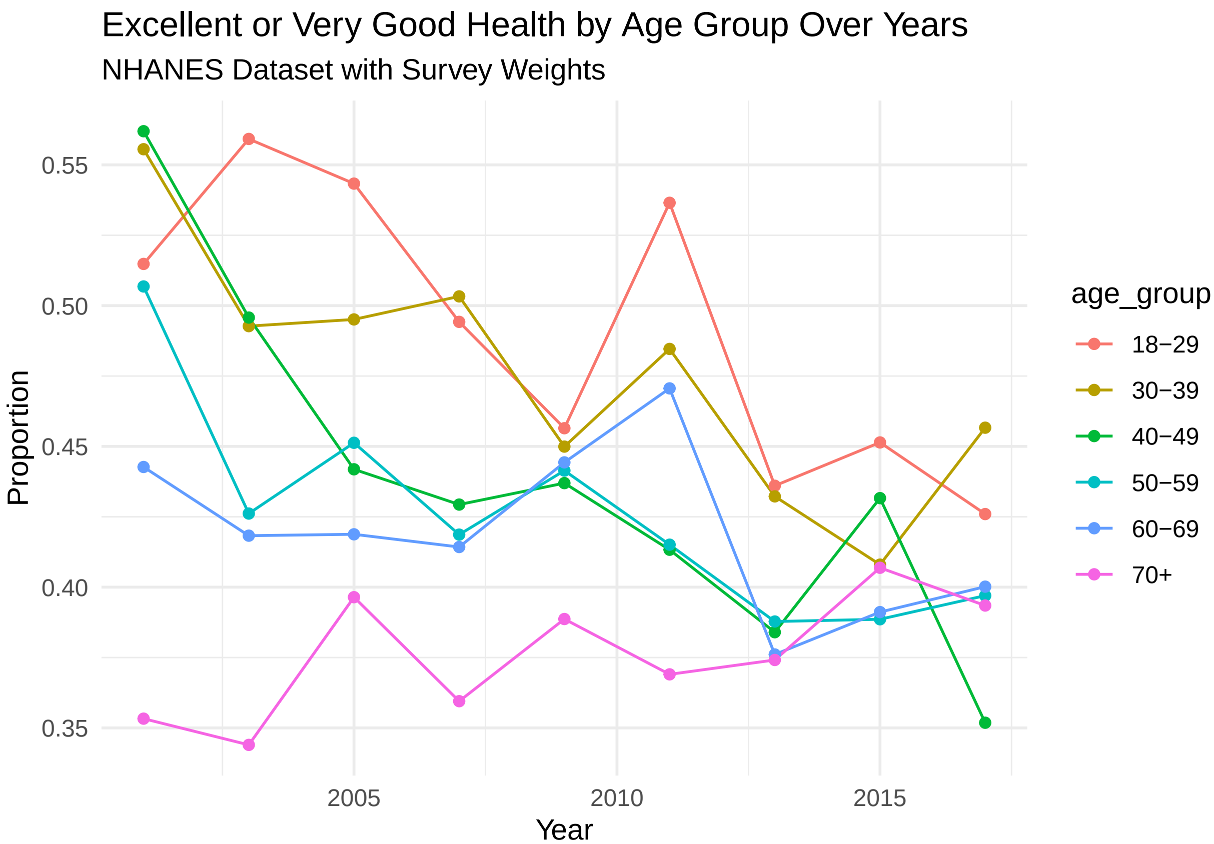


NHANES

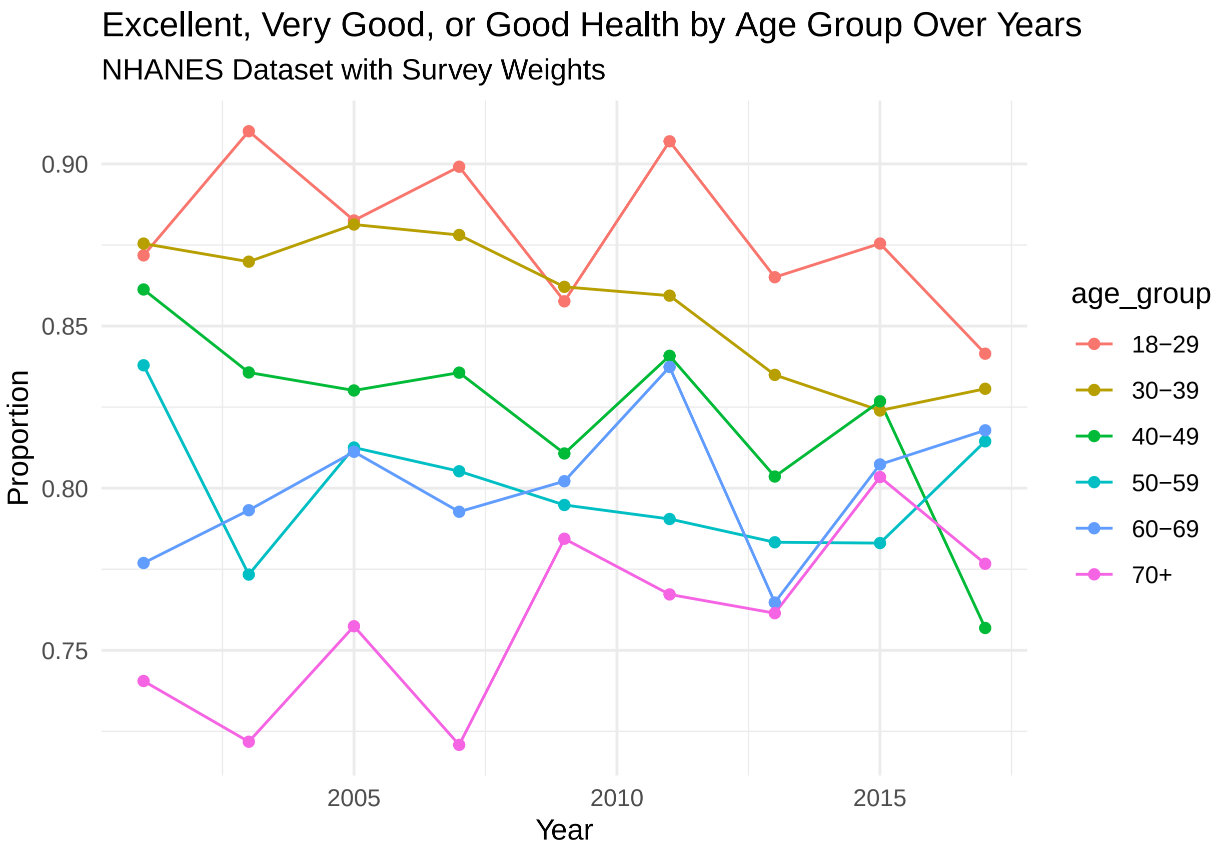
```
p_prop_nhanes_e_vs_vgfp
```



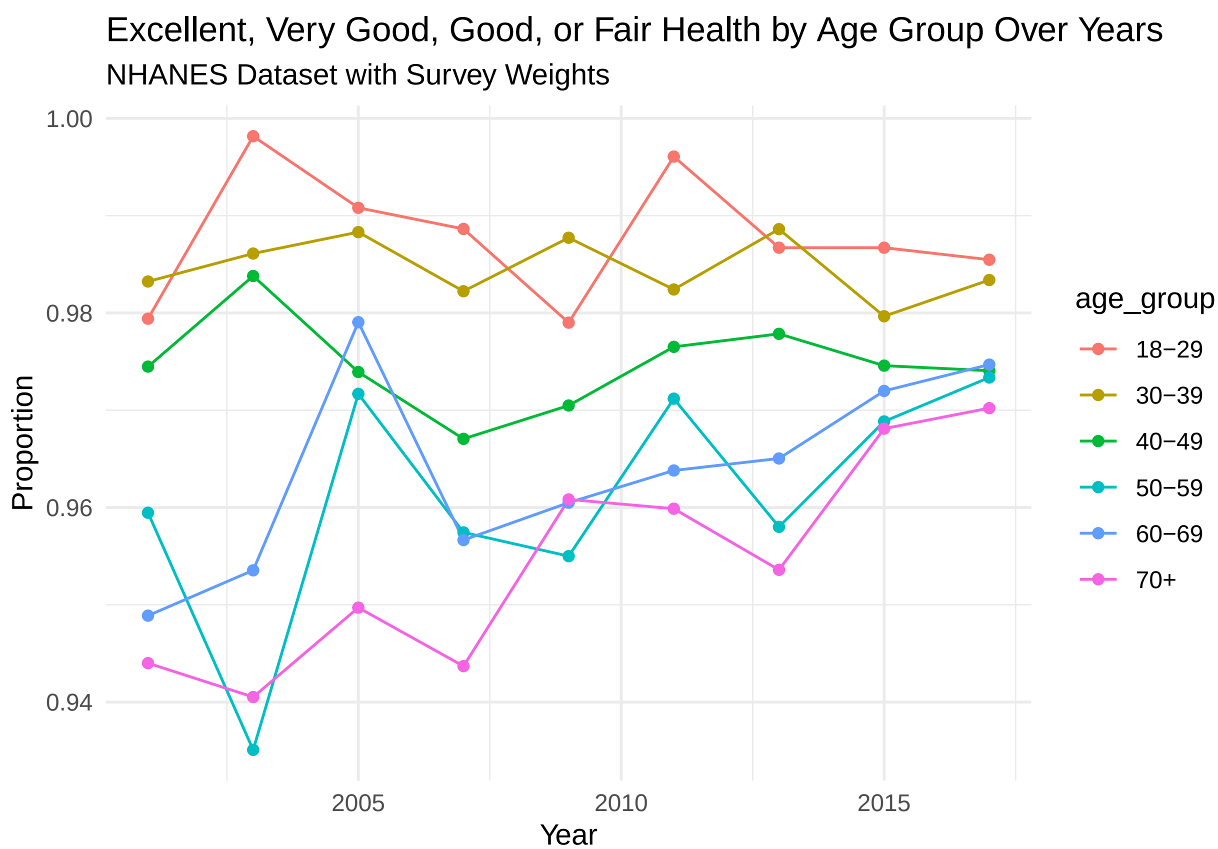

```
p_prop_nhanes_ev_vs_gfp
```




```
p_prop_nhanes_evg_vs_fp
```

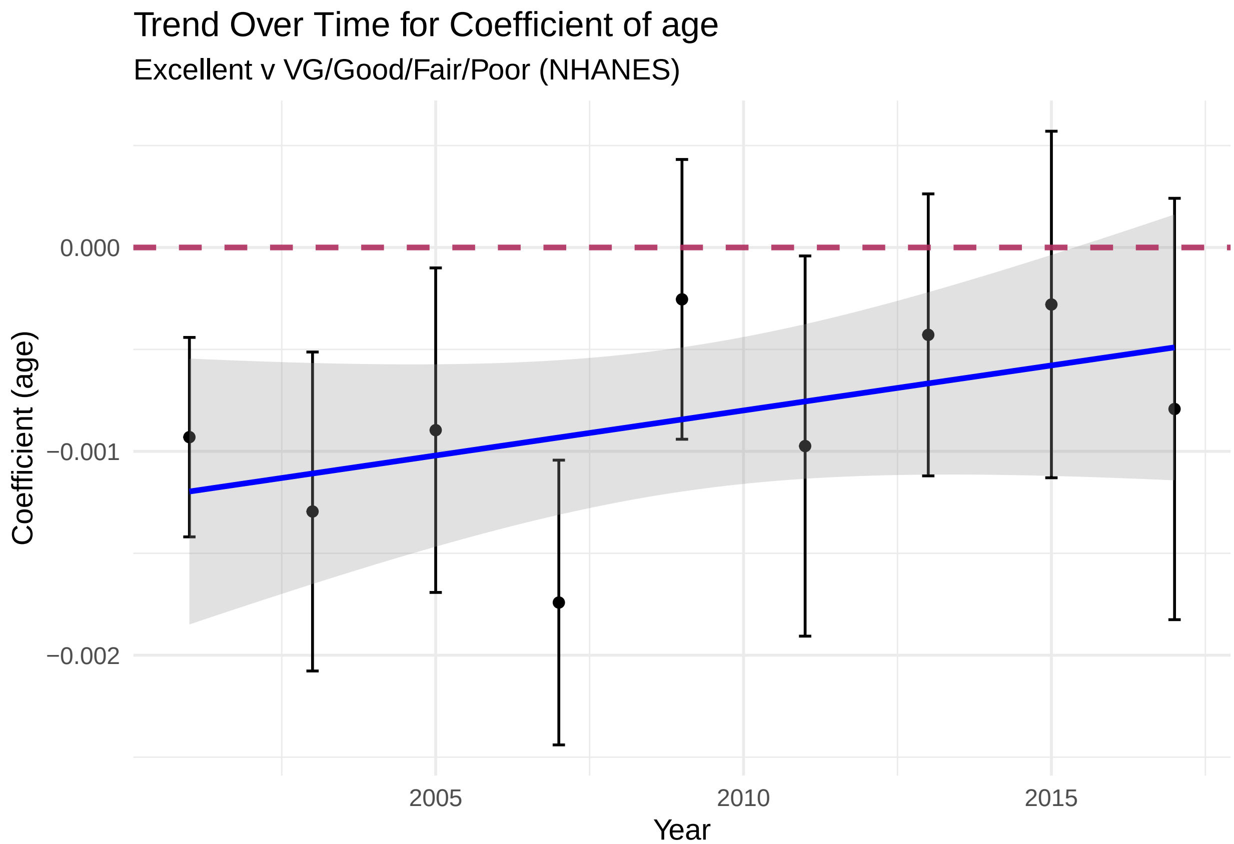



```
p_prop_nhanes_evgf_vs_p
```



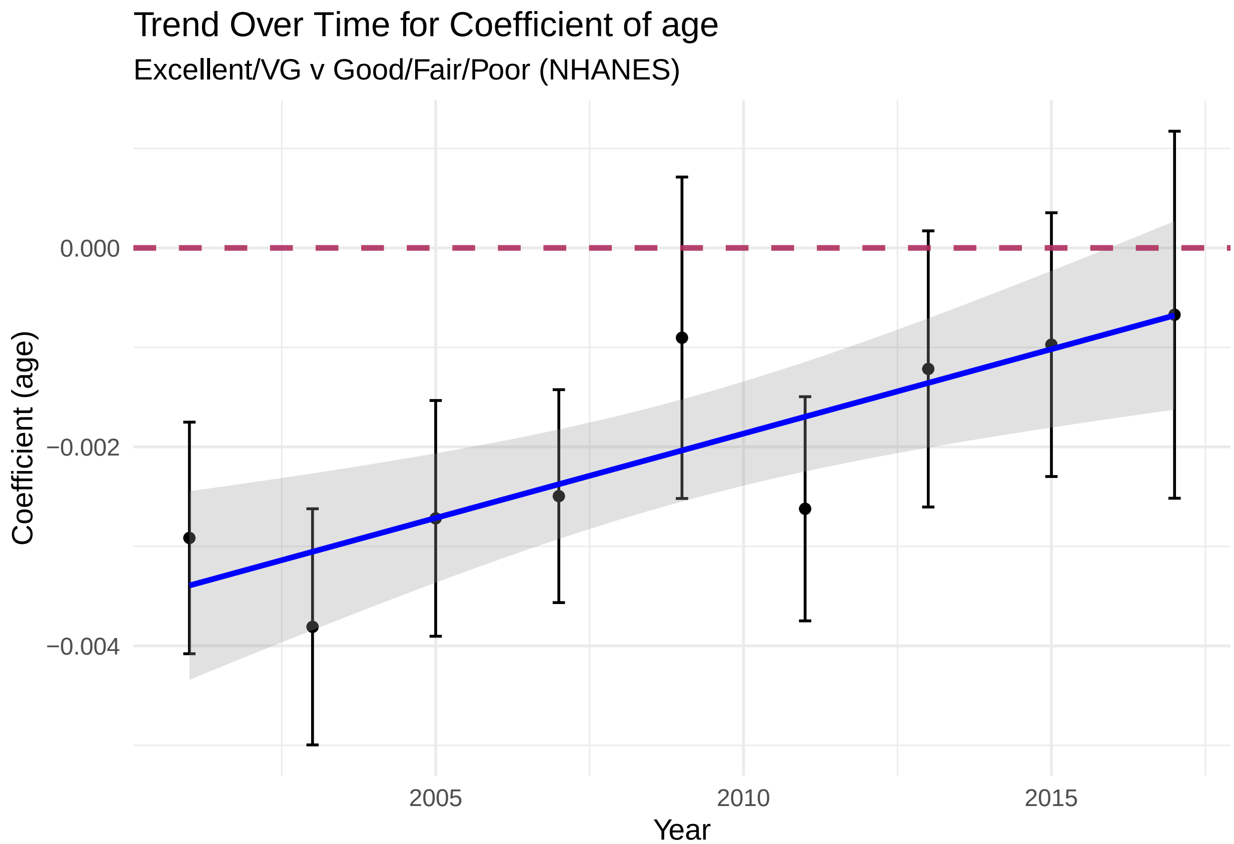

```
p_coeff_nhanes_e_vs_vgfp
```

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



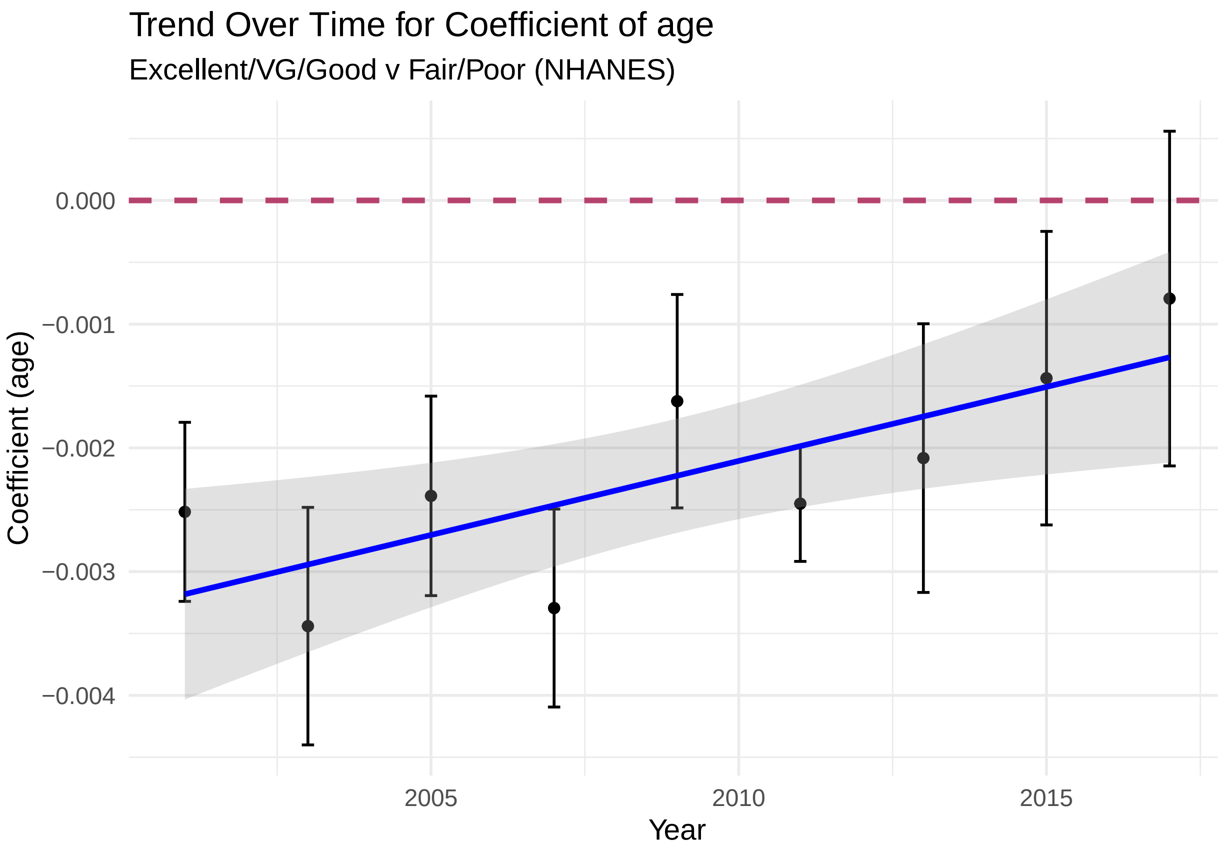

```
p_coeff_nhanes_ev_vs_gfp
```

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



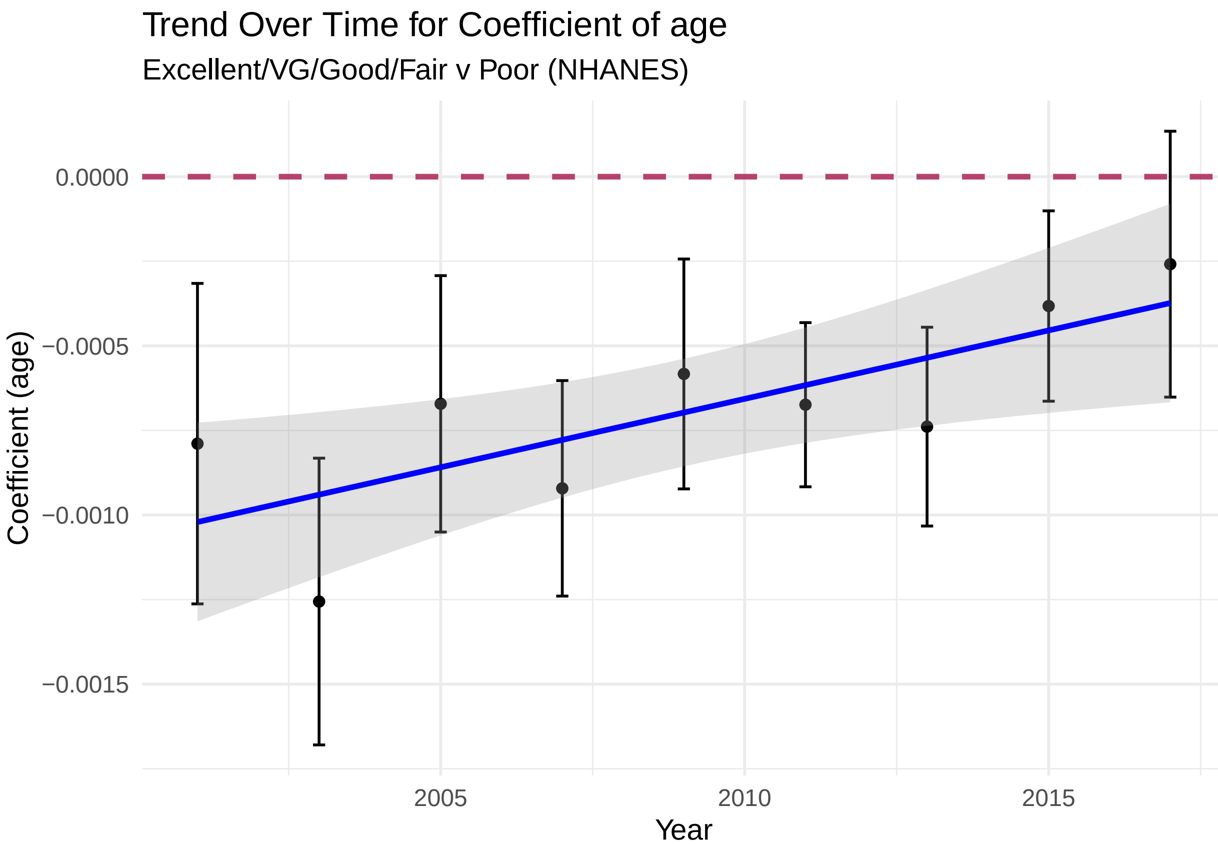

```
p_coeff_nhanes_evg_vs_fp
```

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




```
p_coeff_nhanes_evgf_vs_p
```

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

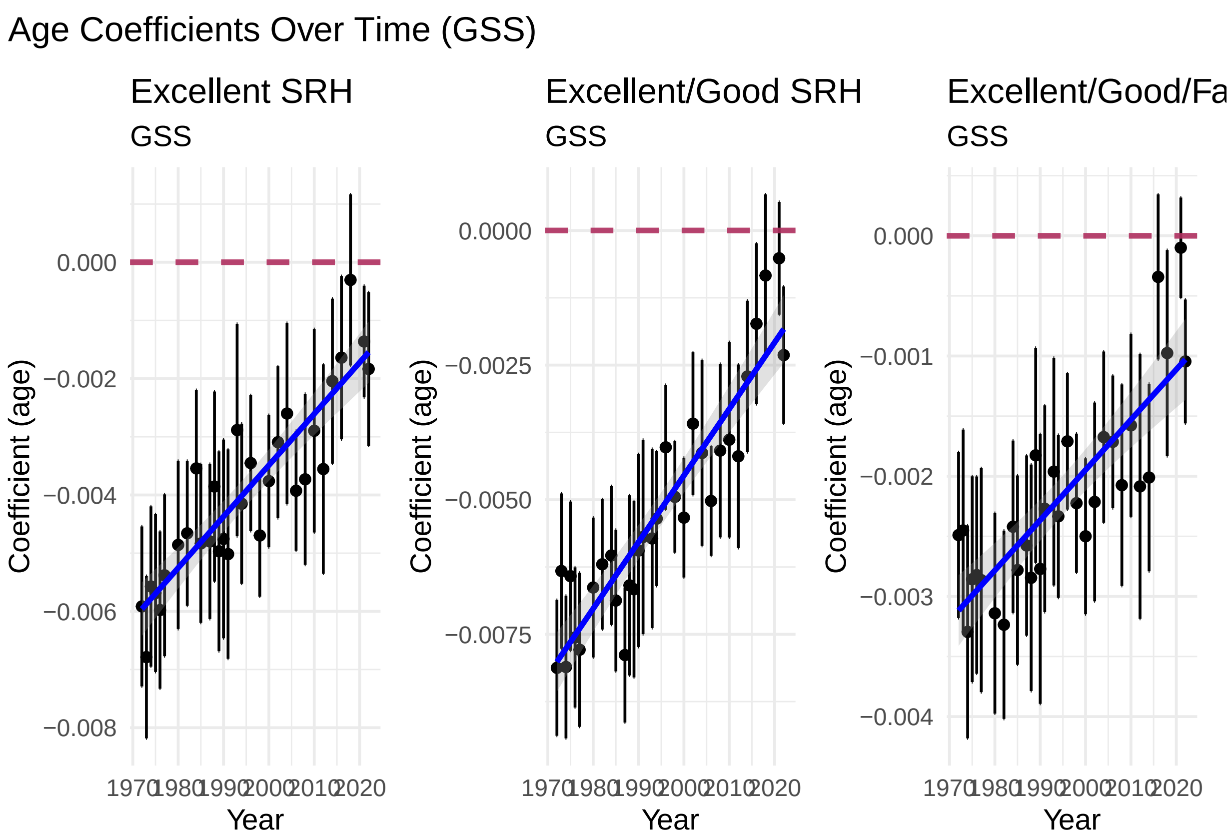


Combined figure

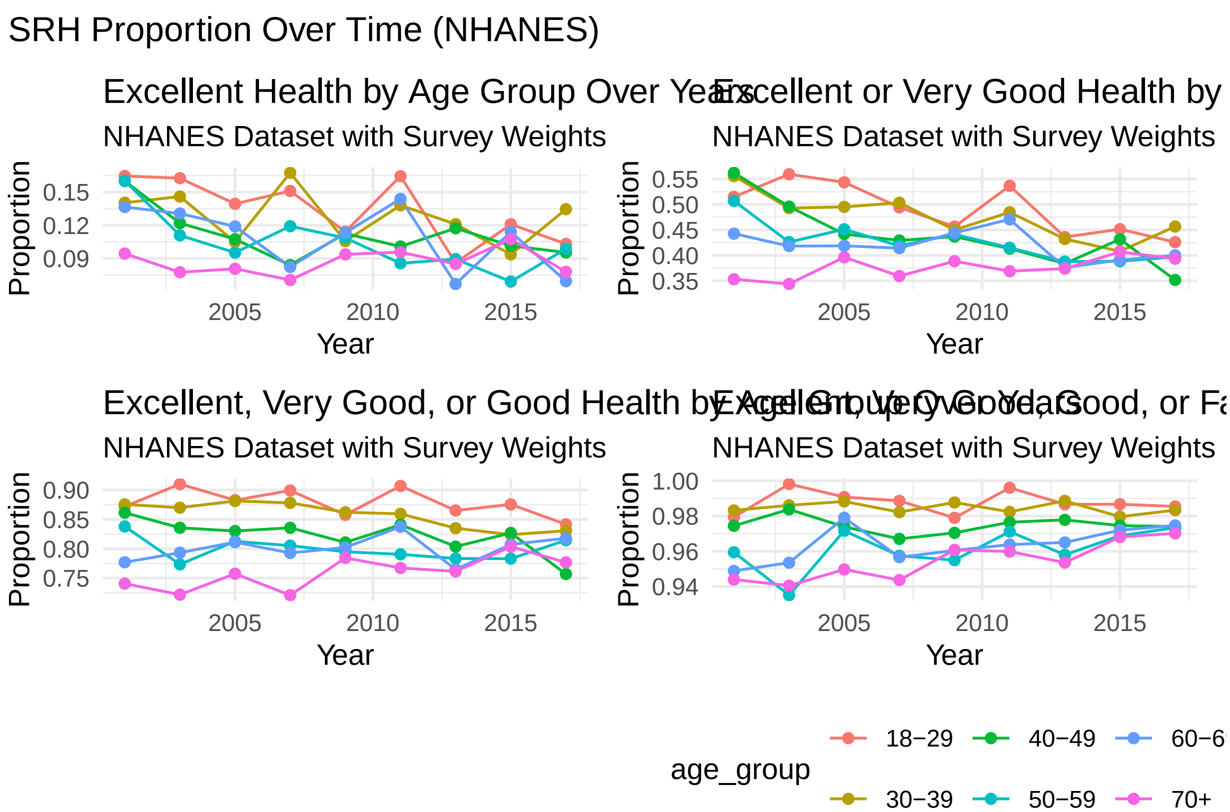
```r
library(patchwork)

# Combine them into a 2x2 grid
# combined_plot <- (p1 + p2) / (p3 + p4)

# or equivalently:
# combined_plot <- p1 + p2 + p3 + p4 +
#                  plot_layout(ncol = 2, nrow = 2)

# Print or save the final combined figure
# combined_plot

# GSS

p_prop_gss_e_vs_gfp + theme(legend.position = "none") + labs(title = "Excellent SRH", subtitle = "GSS")
  p_prop_gss_eg_vs_fp + theme(legend.position = "none") + labs(title = "Excellent/Good SRH", subtitle =
  p_prop_gss_egf_vs_p + labs(title = "Excellent/Good/Fair SRH", subtitle = "GSS") +
  plot_layout(nrow = 1) + plot_annotation("SRH Proportion Over Time (GSS)")
```


```
p_coeff_gss_e_vs_gfp + theme(legend.position = "none") + labs(title = "Excellent SRH", subtitle = "GSS")
  p_coeff_gss_eg_vs_fp + theme(legend.position = "none") + labs(title = "Excellent/Good SRH", subtitle =
  p_coeff_gss_egf_vs_p + labs(title = "Excellent/Good/Fair SRH", subtitle = "GSS") +
  plot_layout(nrow = 1) + plot_annotation("Age Coefficients Over Time (GSS)")
```

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




```r
# NHANES

p_prop_nhanes_e_vs_vgfp +  theme(legend.position = "none") +
  p_prop_nhanes_ev_vs_gfp + theme(legend.position = "none") +
  p_prop_nhanes_evg_vs_fp + theme(legend.position = "none") +
  p_prop_nhanes_evgf_vs_p +  theme(legend.position = "bottom") +
  plot_layout(nrow = 2) + plot_annotation("SRH Proportion Over Time (NHANES)")
```




```r
p_coeff_nhanes_e_vs_vgfp + theme(legend.position = "none") +
p_coeff_nhanes_ev_vs_gfp + theme(legend.position = "none") +
p_coeff_nhanes_evg_vs_fp + theme(legend.position = "none") +
p_coeff_nhanes_evgf_vs_p + theme(legend.position = "bottom") +
  plot_layout(nrow = 2) + plot_annotation("Age Coefficient Over Time (NHANES)")
```

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

Age Coefficient Over Time (NHANES)