NHANES 3 and 4

Christine Lucille Kuryla

2024-12-13

Import and check data

For reference: ### NHANES 3

```
library(SAScii)
# nhanes3.tf <- tempfile()
daturl <- "https://wwwn.cdc.gov/nchs/data/nhanes3/1a/adult.dat"
code_url ="https://wwwn.cdc.gov/nchs/data/nhanes3/1a/adult.sas"
# Sas_code <- url(code_url)
# writeLines ( readLines(Sas_code) , con = nhanes3.tf )
# nhanes3.fwf.parameters <- parse.SAScii( nhanes3.tf , beginline = 5 )
# str( nhanes3.fwf.parameters )
# #-----
# 'data.frame':   90 obs. of  4 variables:
#   $ varname: chr  "SEQN" "HYK1A" "HYK1B" "HYK2A" ...
# $ width  : num  5 1 1 2 2 2 2 4 4 2 ...
# $ char   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
# $ divisor: num  1 1 1 1 1 1 1 1 1 1 ...
# #------

daturl <- "https://wwwn.cdc.gov/nchs/data/nhanes3/1a/adult.dat"
in.nhanes3 <- read.fwf(daturl, widths=nhanes3.fwf.parameters$width,
                     col.names= nhanes3.fwf.parameters$varname)

in2 <- read.SAScii( daturl, code_url)

#write_csv(in2, "big_data/NHANES/nhanes_3/nhanes3.csv")

nhanes3_data <- read_csv("big_data/NHANES/nhanes_3/nhanes3.csv")

nhanes3_selected <- nhanes3_data %>%
  select(SEQN,
         DMPFSEQ,
         HSAGEIR, # age in years
         HAB1, # self-rated health: 1:excellent, very good, good, fair, 5: poor (get rid of 6 and 7)
         HSSEX, # 1 male, 2 female
         SDPPHASE, # 1 1988-1991, 2 1991-1994
         HSDOIMO, # date of screener (month)
         HSAGEU, # age unit
         HSAITMOR # age in months at interview (screener)
         ) %>%
  filter(HAB1 %in% 1:5) %>%
```

```r
  mutate(age = HSAGEIR,
         sex = ifelse(HSSEX == 1, "Male", "Female"),
         year = ifelse(SDPPHASE == 1, 1989.5, 1992.5),
         srh = 6 - HAB1)

glimpse(nhanes3_selected)

#write_csv(nhanes3_selected, "data/nhanes3_selected.csv")
```

NHANES 4

```r
nhanes4_key <- read_csv("big_data/NHANES/nhanes_4/nhanes4_key.csv")

library(tidyverse)
library(haven)

# Assume nhanes4_key is loaded

# Separate keys by type of file (DEMO, HUQ) for simplicity
demo_key <- nhanes4_key %>% filter(str_detect(nhanes_file, "^DEMO"))
huq_key <- nhanes4_key %>% filter(str_detect(nhanes_file, "^HUQ"))

# A helper function to read and process a given domain of files
read_nhanes_domain <- function(key_table) {
  # Get unique files for this domain
  files <- key_table %>% distinct(nhanes_file)

  domain_data <- files %>%
    mutate(
      data = map(nhanes_file, ~ {
        vars_for_file <- key_table %>% filter(nhanes_file == .x)
        needed_vars <- c("SEQN", unique(vars_for_file$nhanes_var))

        file_path <- paste0("big_data/NHANES/nhanes_4/", .x, ".xpt")

        # Read and select needed variables
        df <- read_xpt(file_path) %>%
          select(any_of(needed_vars)) %>%
          # Rename nhanes_var to my_var
          rename_with(
            .fn = ~ vars_for_file$my_var[match(., vars_for_file$nhanes_var)],
            .cols = vars_for_file$nhanes_var
          ) %>%
          mutate(
            nhanes_yr_1 = vars_for_file$nhanes_yr_1[1],
            nhanes_yr_2 = vars_for_file$nhanes_yr_2[1]
          )

        df
      })
    ) %>%
    unnest(cols = data)  # Unnest after renaming done
  domain_data
}


# Read DEMO and HUQ data separately
demo_data <- read_nhanes_domain(demo_key)
huq_data <- read_nhanes_domain(huq_key)

# Now join demo and huq data by SEQN and cycle years.
# Note: If multiple cycles overlap, you may need to use both SEQN and nhanes_yr_1/nhanes_yr_2 as join k
# Typically SEQN is unique within a cycle, so joining on SEQN and year information might be prudent.
final_data <- demo_data %>%
  full_join(huq_data, by = c("SEQN", "nhanes_yr_1", "nhanes_yr_2"))

# Now select the columns you need:
final_data <- final_data %>%
  select(
    SEQN,
    age,
    srh_huq010,
    SDDSRVYR,
    nhanes_yr_1,
    nhanes_yr_2
  )

glimpse(final_data)

nhanes4_selected <- final_data %>%
  filter(srh_huq010 %in% 1:5) %>%
  filter(age >= 18) %>%
  mutate(srh = 6 - srh_huq010) %>%
  mutate(year = (nhanes_yr_1 + nhanes_yr_2 ) / 2 ) %>%
  mutate(cohort = year - age)

glimpse(nhanes4_selected)

write_csv(nhanes4_selected, "big_data/NHANES/nhanes_4/nhanes4_selected_apcsrh.csv")
write_csv(nhanes4_selected, "data/nhanes4_selected_apcsrh.csv")
```

Import and check formatted data

```r
nhanes3 <- read_csv("data/nhanes3_selected.csv") %>%
  select(SEQN, age, year, cohort, srh)
```

```
## Rows: 20037 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): sex
## dbl (13): SEQN, DMPFSEQ, HSAGEIR, HAB1, HSSEX, SDPPHASE, HSDOIMO, HSAGEU, HS...
##
## 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
nhanes4 <- read_csv("data/nhanes4_selected_apcsrh.csv") %>%
  select(SEQN, age, year, cohort, srh)
```

```
## Rows: 113188 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): SEQN, age, srh_huq010, SDDSRVYR, nhanes_yr_1, nhanes_yr_2, srh, yea...
##
## 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 <- rbind(nhanes3, nhanes4)

data_nhanes <- data_nhanes %>%
  na.omit() %>%
  filter(age >= 18)

glimpse(data_nhanes)
```

```
## Rows: 117,441
## Columns: 5
## $ SEQN   <dbl> 3, 4, 9, 10, 11, 19, 34, 44, 45, 48, 49, 51, 52, 53, 54, 55, 56~
## $ age    <dbl> 21, 32, 48, 35, 48, 44, 42, 24, 67, 56, 82, 44, 50, 36, 19, 48,~
## $ year   <dbl> 1989.5, 1989.5, 1989.5, 1989.5, 1989.5, 1989.5, 1989.5, 1989.5,~
## $ cohort <dbl> 1968.5, 1957.5, 1941.5, 1954.5, 1941.5, 1945.5, 1947.5, 1965.5,~
## $ srh    <dbl> 5, 4, 4, 4, 2, 4, 5, 3, 4, 4, 3, 3, 5, 3, 3, 2, 4, 3, 2, 3, 4, ~
```

```r
table(data_nhanes$srh)
```

```
##
##     1     2     3     4     5
##  3968 17146 36570 29868 29889
```

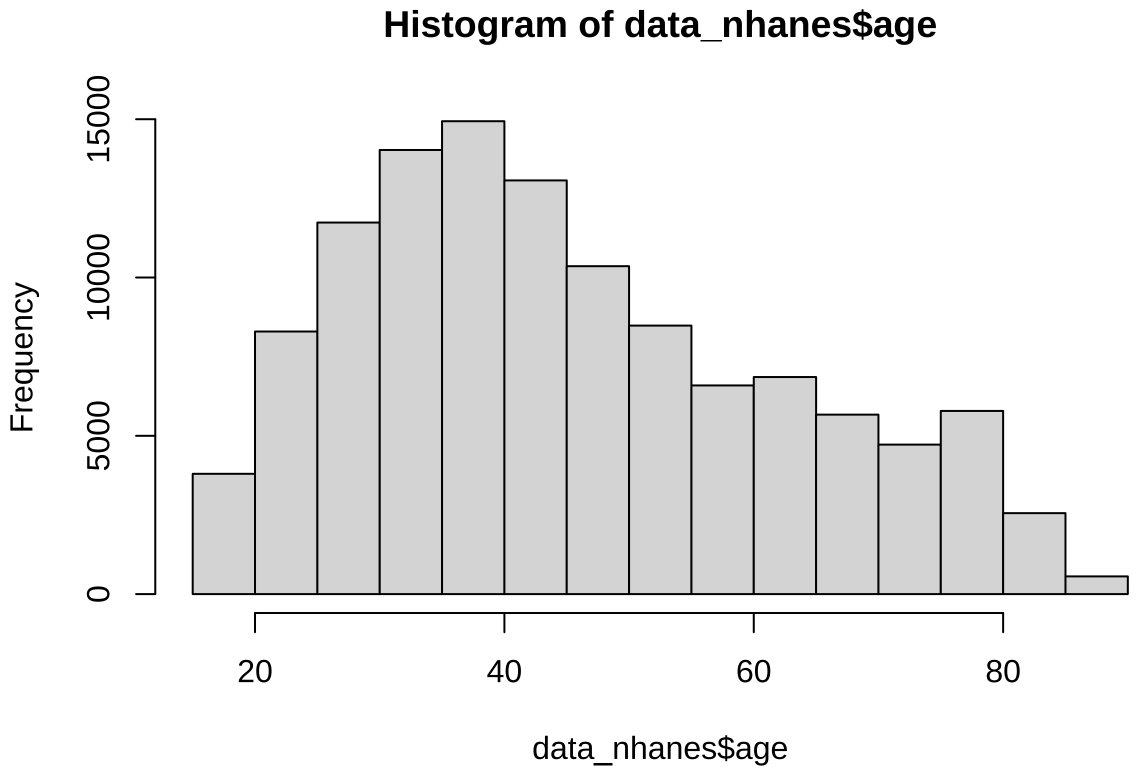
```r
table(data_nhanes$year)
```

```
##
## 1989.5 1992.5 1999.5 2001.5 2003.5 2005.5 2007.5 2009.5 2011.5 2013.5 2015.5
##   9890   9715   9942  11026  10114  10340  10146  10534   9754  10170   9961
## 2017.5
##   5849
```

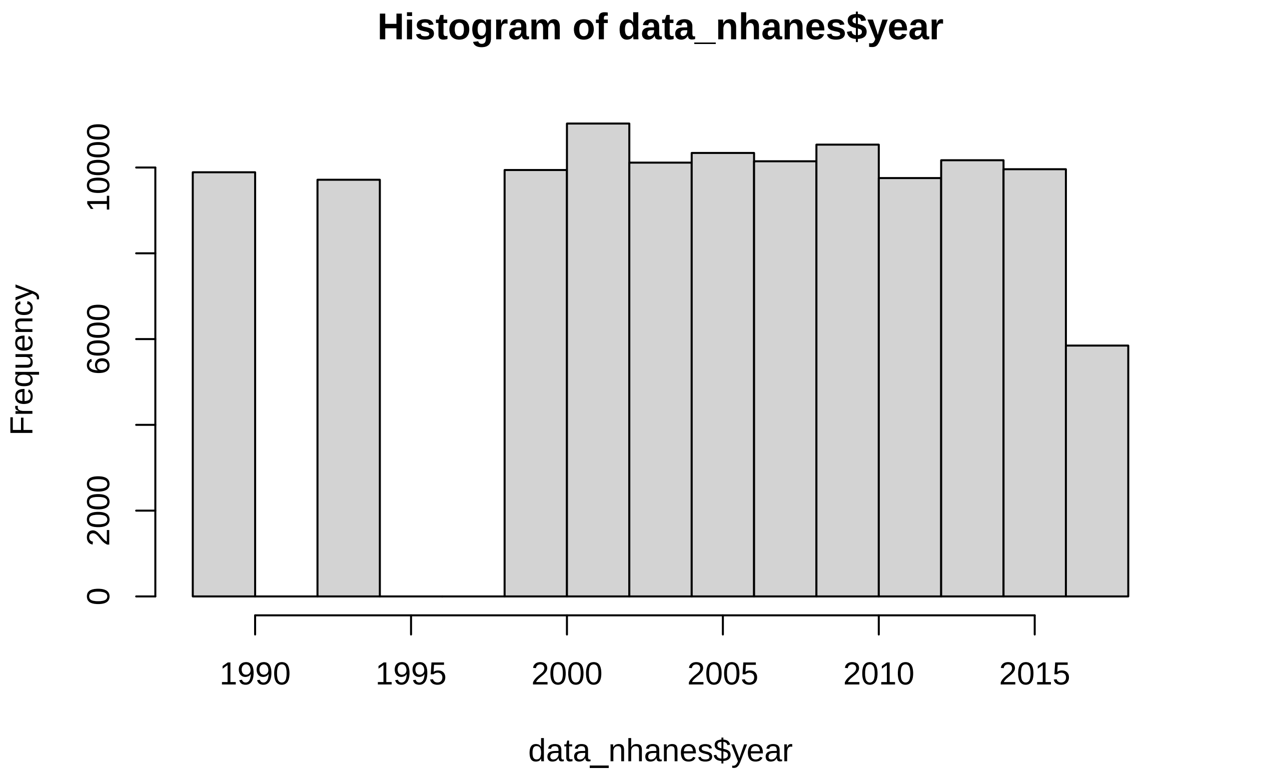
```r
table(data_nhanes$cohort)
```

```
##
## 1899.5 1900.5 1901.5 1902.5 1903.5 1904.5 1905.5 1906.5 1907.5 1908.5 1909.5
##    104     35     38    155     92    103    150    169    179    222    226
## 1910.5 1911.5 1912.5 1913.5 1914.5 1915.5 1916.5 1917.5 1918.5 1919.5 1920.5
##    174    201    240    170    324    233    484    350    554    447    603
## 1921.5 1922.5 1923.5 1924.5 1925.5 1926.5 1927.5 1928.5 1929.5 1930.5 1931.5
##    476    449    467    501    503    511    993    594   1041    564   1058
## 1932.5 1933.5 1934.5 1935.5 1936.5 1937.5 1938.5 1939.5 1940.5 1941.5 1942.5
##    734    986    782   1202    823   1336    933   1067   1007   1010   1152
## 1943.5 1944.5 1945.5 1946.5 1947.5 1948.5 1949.5 1950.5 1951.5 1952.5 1953.5
##   1195   1067   1195   1326   1502   1501   1573   1549   1676   1670   1858
## 1954.5 1955.5 1956.5 1957.5 1958.5 1959.5 1960.5 1961.5 1962.5 1963.5 1964.5
##   1978   2107   2278   2253   2366   2424   2495   2545   2598   2708   2665
## 1965.5 1966.5 1967.5 1968.5 1969.5 1970.5 1971.5 1972.5 1973.5 1974.5 1975.5
##   2602   2566   2703   2762   2891   2899   2939   2412   2416   2284   2173
## 1976.5 1977.5 1978.5 1979.5 1980.5 1981.5 1982.5 1983.5 1984.5 1985.5 1986.5
##   2078   2099   1864   1966   1870   1880   1435   1320   1377   1146   1012
## 1987.5 1988.5 1989.5 1990.5 1991.5 1992.5 1993.5 1994.5 1995.5 1996.5 1997.5
##    878    710    649    523    448    356    322    211    190    109     93
## 1998.5 1999.5
##    143    144
```

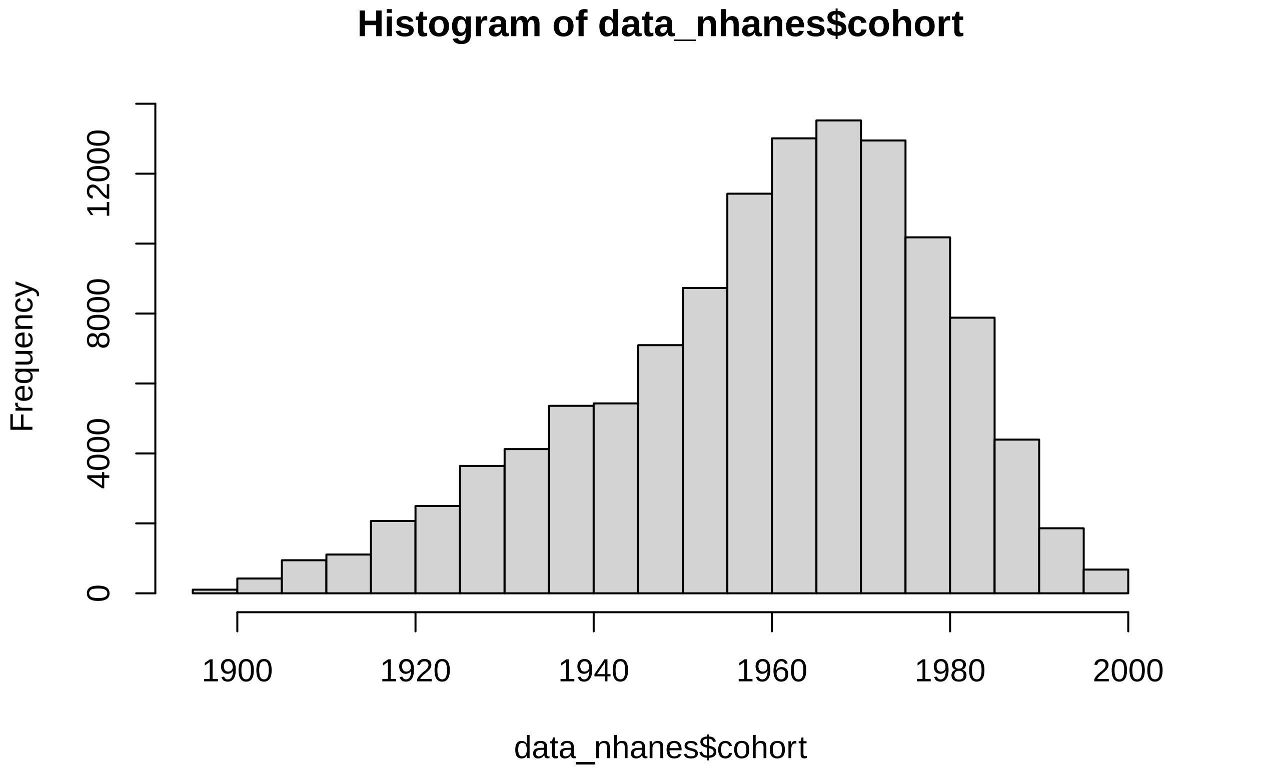
```
hist(data_nhanes$age)
```



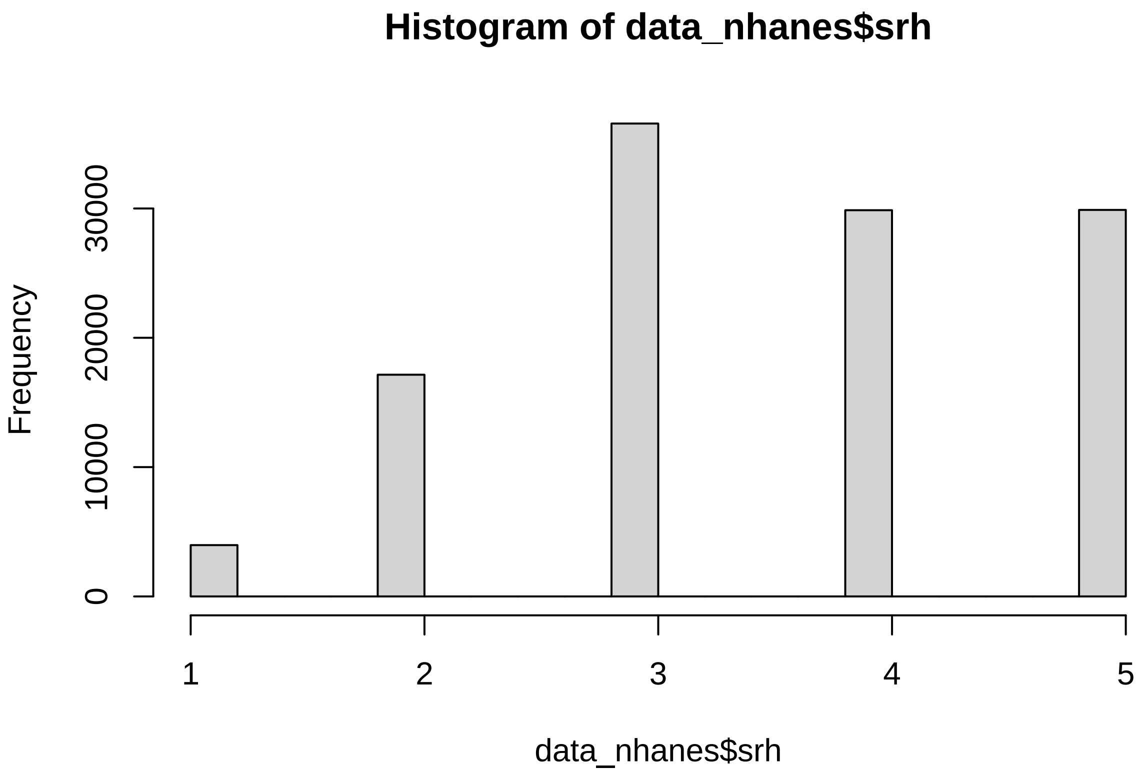
```
hist(data_nhanes$year)
```



```r
hist(data_nhanes$cohort)
```

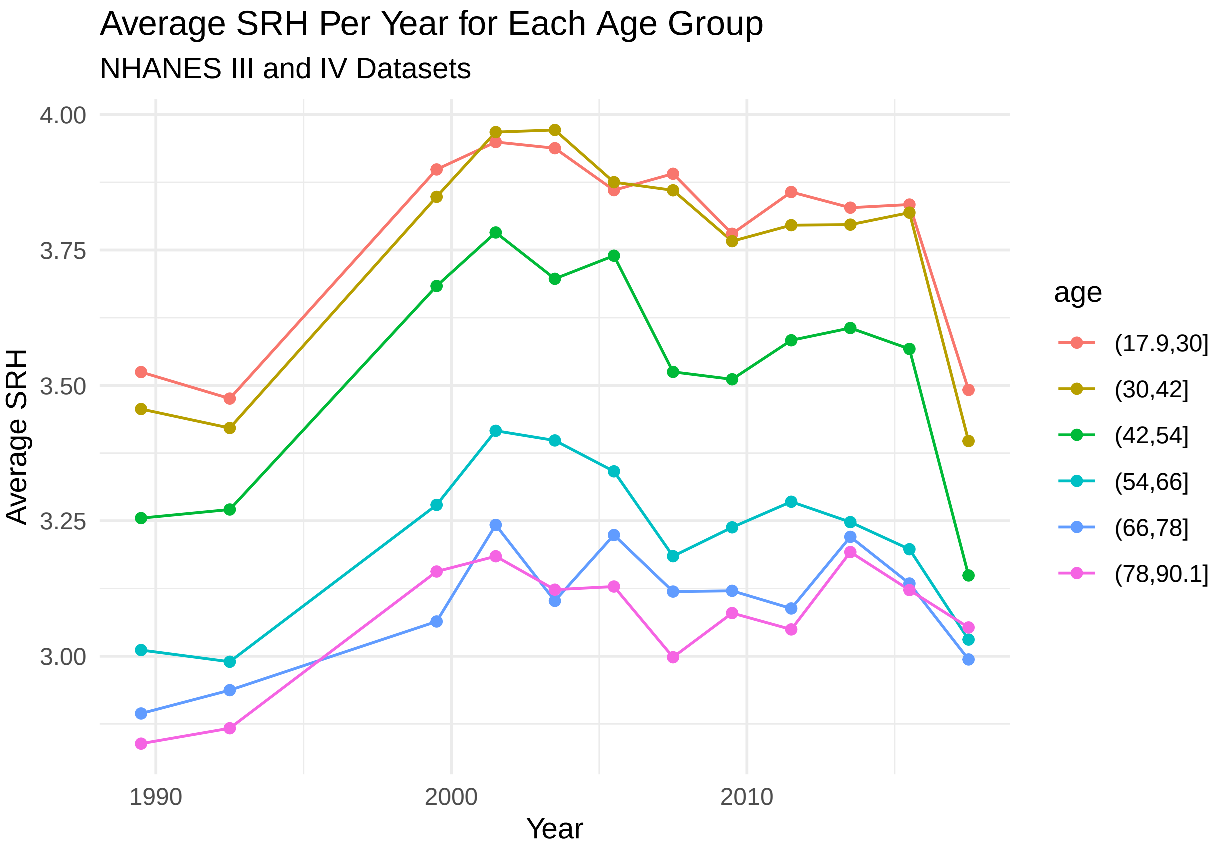


```r
hist(data_nhanes$srh)
```



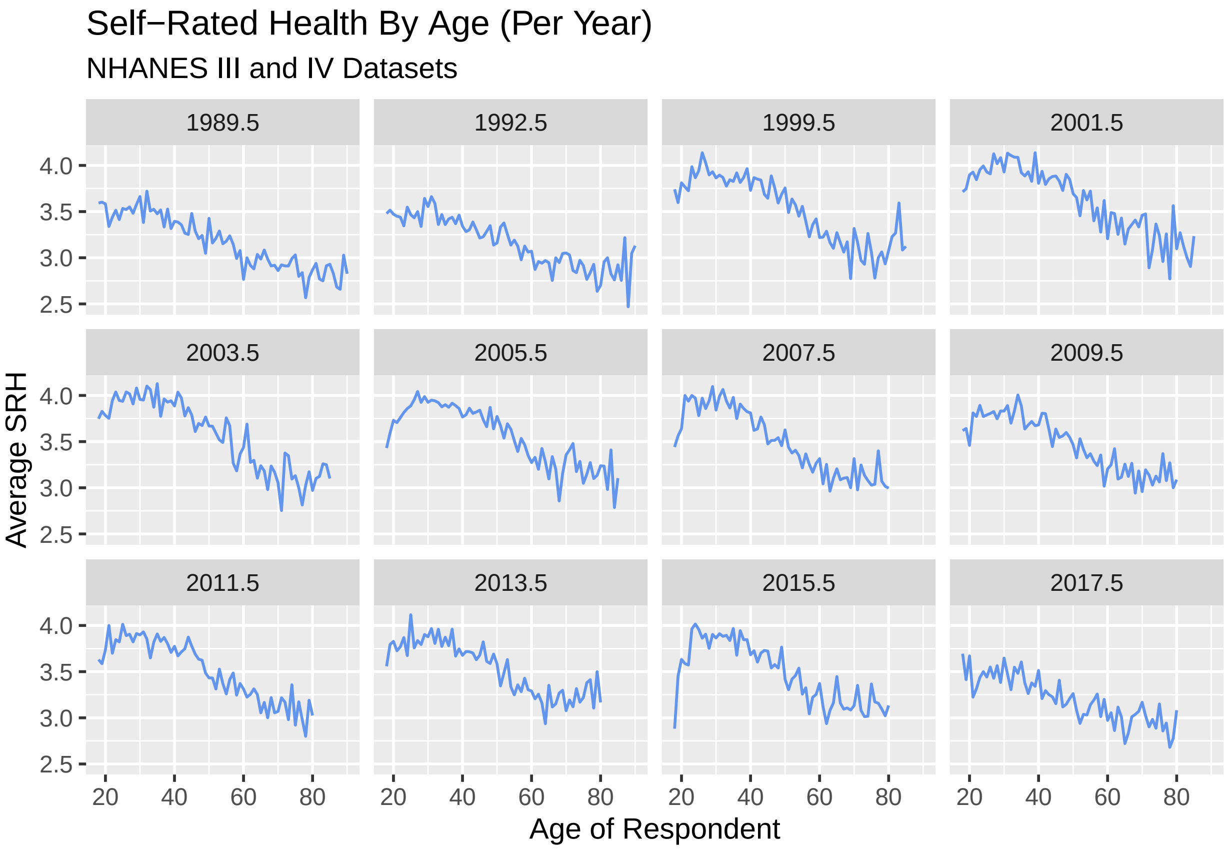
NHANES III and IV

```
data_nhanes %>%
    mutate(age = cut(age, breaks = 6)) %>% # Create cohorts with 6 breaks
    group_by(age, year) %>%
    dplyr::summarize(mean_health = mean(srh, na.rm = TRUE), .groups = "drop") %>%
    ggplot(aes(x = year, y = mean_health, color = age)) +
    geom_line() +
    geom_point() +
    theme_minimal() +
    labs(title = "Average SRH Per Year for Each Age Group",
         subtitle = "NHANES III and IV Datasets",
         x = "Year",
         y = "Average SRH")
```



```
data_nhanes %>%
  group_by(age, year) %>%
  summarize(mean_health = mean(srh)) %>%
  ggplot(aes(x = age, y = mean_health)) +
  geom_line(color = "cornflowerblue") +
  facet_wrap(~ year) +
  labs(title = "Self-Rated Health By Age (Per Year)",
       subtitle = "NHANES III and IV Datasets",
       x = "Age of Respondent",
       y = "Average SRH",
       )
```

```
## `summarise()` has grouped output by 'age'. You can override using the `.groups`
## argument.
```



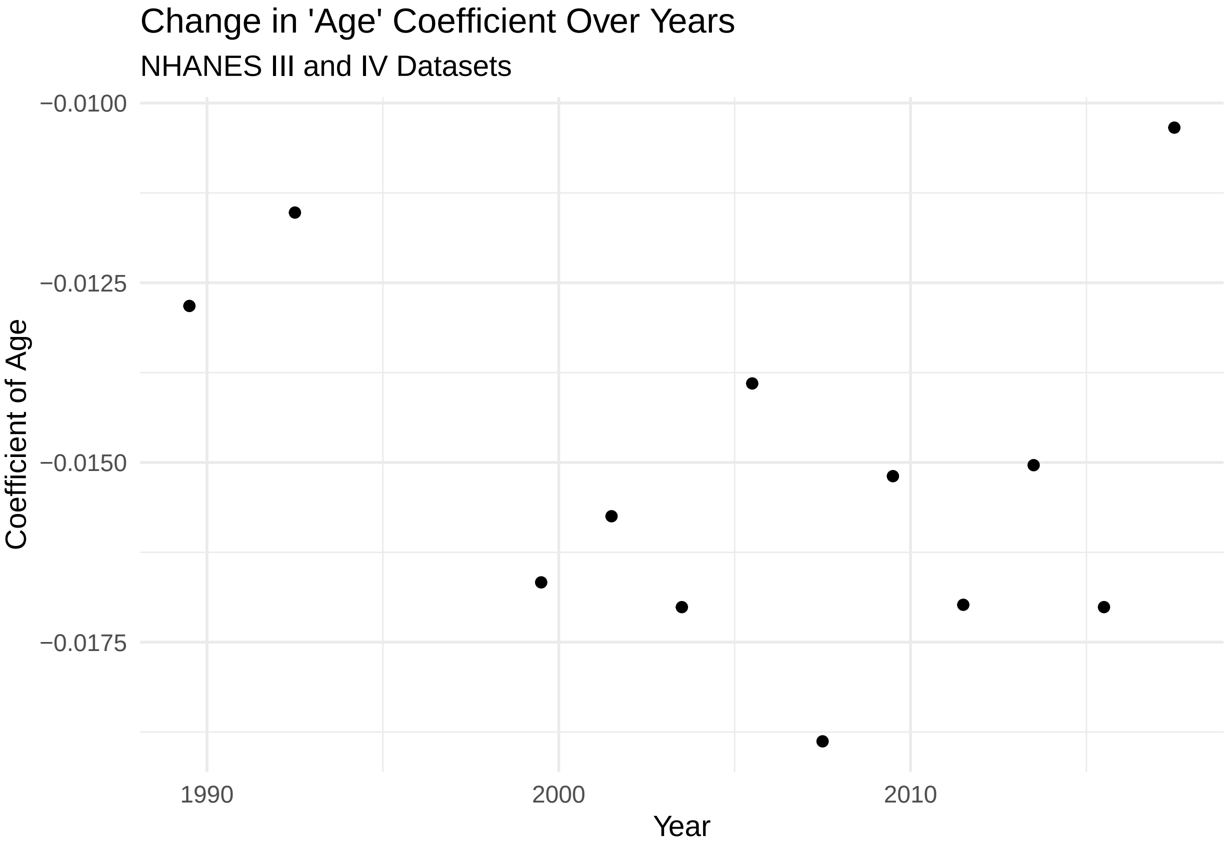
```r
library(broom)

# Aggregate slopes

# Perform linear regression for each year and extract the coefficient of 'age' with confidence interval
lm_health_v_age_0 <- data_nhanes %>%
  group_by(year) %>%
  do(tidy(lm(srh ~ age, data = .), conf.int = TRUE)) %>%  # Add conf.int = TRUE for CIs
  filter(term == "age") %>%
  select(year, coef = estimate, conf.low, conf.high, se = std.error, t_statistic = statistic,  p_value

# View the results with confidence intervals, se, t statistic, and p value
# print(lm_health_v_age_0)
knitr::kable(lm_health_v_age_0,
             caption = "NHANES III and IV Datasets")
```

Table 1: NHANES III and IV Datasets

year	coef	conf.low	conf.high	se	t_statistic	p_value
1989.5	-0.0128228	-0.0138368	-0.0118089	0.0005173	-24.79012	0
1992.5	-0.0115236	-0.0125728	-0.0104745	0.0005352	-21.53062	0
1999.5	-0.0166679	-0.0179948	-0.0153411	0.0006769	-24.62424	0
2001.5	-0.0157482	-0.0169775	-0.0145189	0.0006271	-25.11140	0
2003.5	-0.0170125	-0.0182480	-0.0157771	0.0006303	-26.99218	0
2005.5	-0.0139011	-0.0151836	-0.0126185	0.0006543	-21.24532	0
2007.5	-0.0188795	-0.0201763	-0.0175826	0.0006616	-28.53685	0
2009.5	-0.0151905	-0.0164678	-0.0139132	0.0006516	-23.31204	0

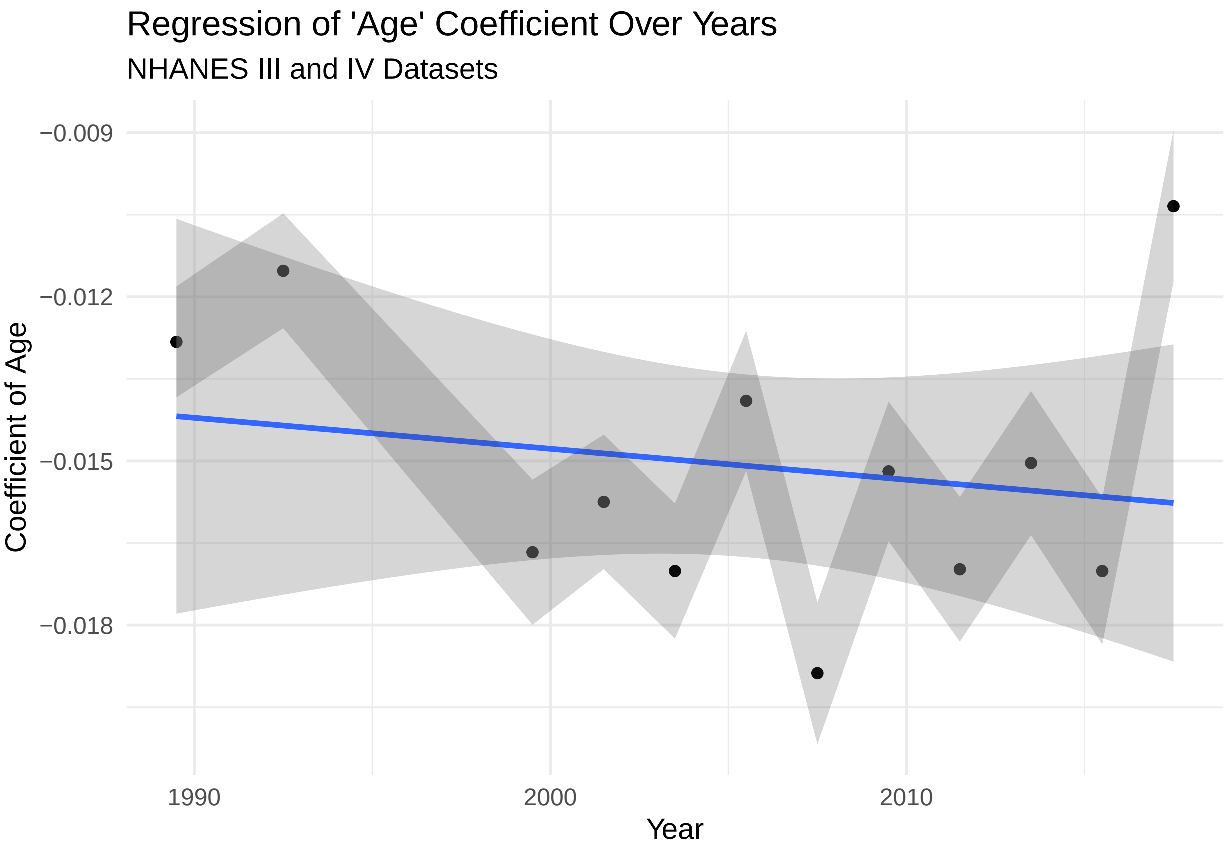
year	coef	conf.low	conf.high	se	t_statistic	p_value
2011.5	-0.0169800	-0.0183033	-0.0156567	0.0006751	-25.15261	0
2013.5	-0.0150377	-0.0163551	-0.0137203	0.0006721	-22.37523	0
2015.5	-0.0170124	-0.0183518	-0.0156729	0.0006833	-24.89712	0
2017.5	-0.0103421	-0.0117272	-0.0089570	0.0007066	-14.63698	0

```r
# Plot coefficients
ggplot(lm_health_v_age_0, aes(x = year, y = coef)) +
  geom_point() +
  labs(
    title = "Change in 'Age' Coefficient Over Years",
    subtitle = "NHANES III and IV Datasets",
    x = "Year",
    y = "Coefficient of Age"
  ) +
  theme_minimal()
```



```r
## Regress the srh vs age coefficients from each year on the year of the survey


# Visualize
ggplot(lm_health_v_age_0, aes(x = year, y = coef)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +  # Adds the regression line with standard error shading
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.2) +  # Confidence intervals for the co
  labs(
    title = "Regression of 'Age' Coefficient Over Years",
    subtitle = "NHANES III and IV Datasets",
    x = "Year",
    y = "Coefficient of Age"
  ) +
  theme_minimal()
```

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

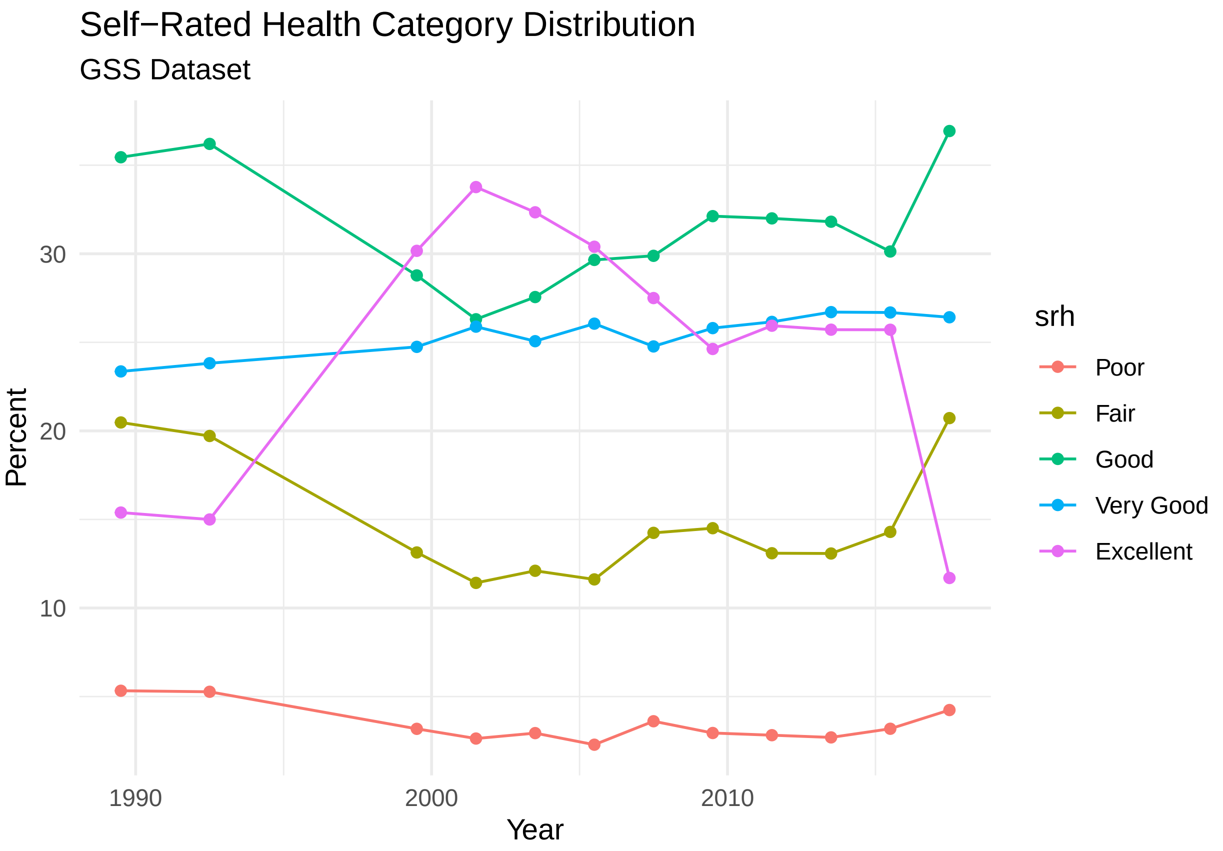


```
# Perform linear regression of 'coef' (age coefficient) vs 'year'
lm_coef_vs_year <- lm(coef ~ year, data = lm_health_v_age_0)

# View the summary of the regression
summary(lm_coef_vs_year)
```

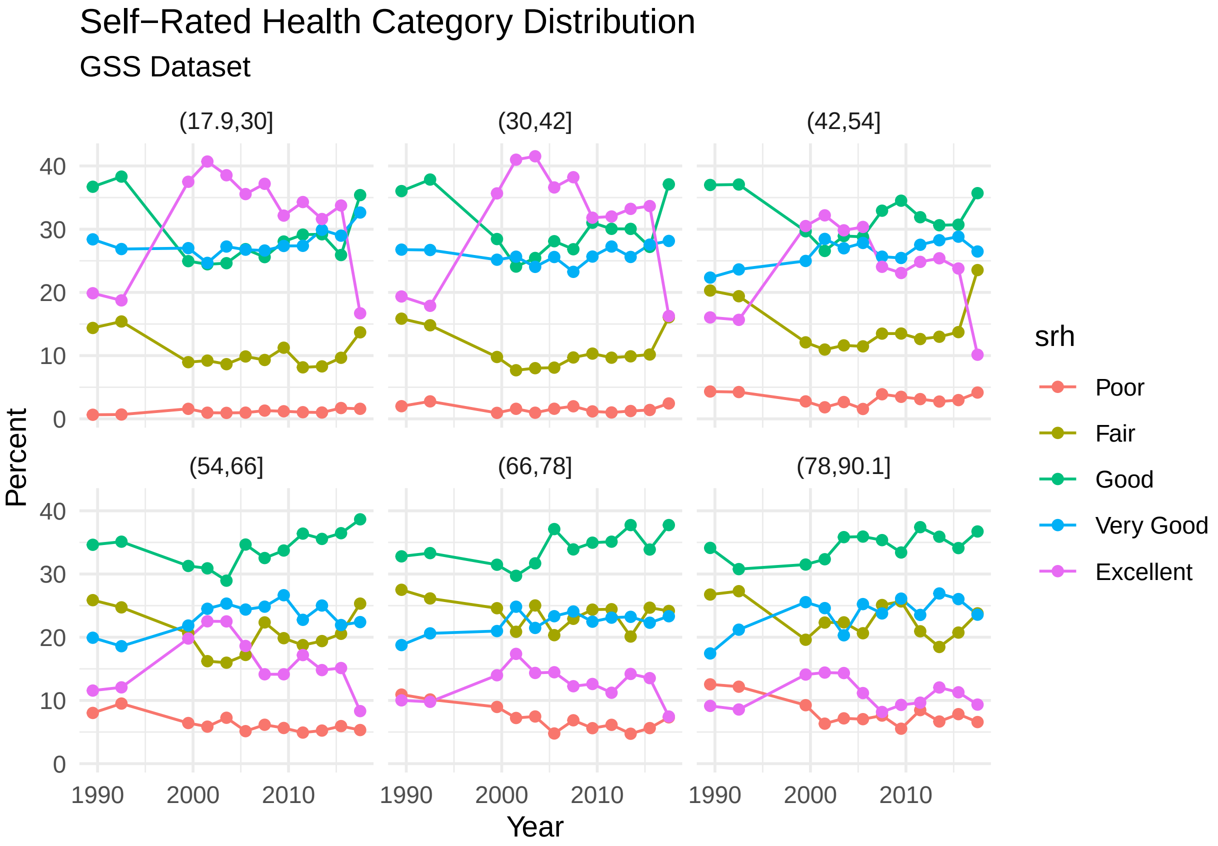
```
##
## Call:
## lm(formula = coef ~ year, data = lm_health_v_age_0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0036777 -0.0016436 -0.0003809  0.0012305  0.0054258
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.847e-02  1.791e-01   0.550    0.595
## year        -5.662e-05  8.929e-05  -0.634    0.540
##
```

```
## Residual standard error: 0.002595 on 10 degrees of freedom
## Multiple R-squared:  0.03866,    Adjusted R-squared:  -0.05748
## F-statistic: 0.4021 on 1 and 10 DF,  p-value: 0.5402
```

```
# Self-Rated Health Category Distribution
data_nhanes %>%
  mutate(health = srh) %>%
  select(year, health) %>%
  filter(!is.na(health)) %>%
  mutate(
    srh = factor(health,
                 levels = 1:5,
                 labels = c("Poor", "Fair", "Good", "Very Good", "Excellent"))) %>%
  # Remove missing values
  # Calculate percentages by year
  group_by(year) %>%
  count(srh) %>%
  mutate(percent = n / sum(n) * 100) %>%
  ungroup() %>%
  ggplot(aes(x = year, y = percent, color = srh)) +
  geom_line() +
  geom_point() +
  labs(title = "Self-Rated Health Category Distribution",
       subtitle = "GSS Dataset",
       y = "Percent",
       x = "Year") +
  theme_minimal()
```



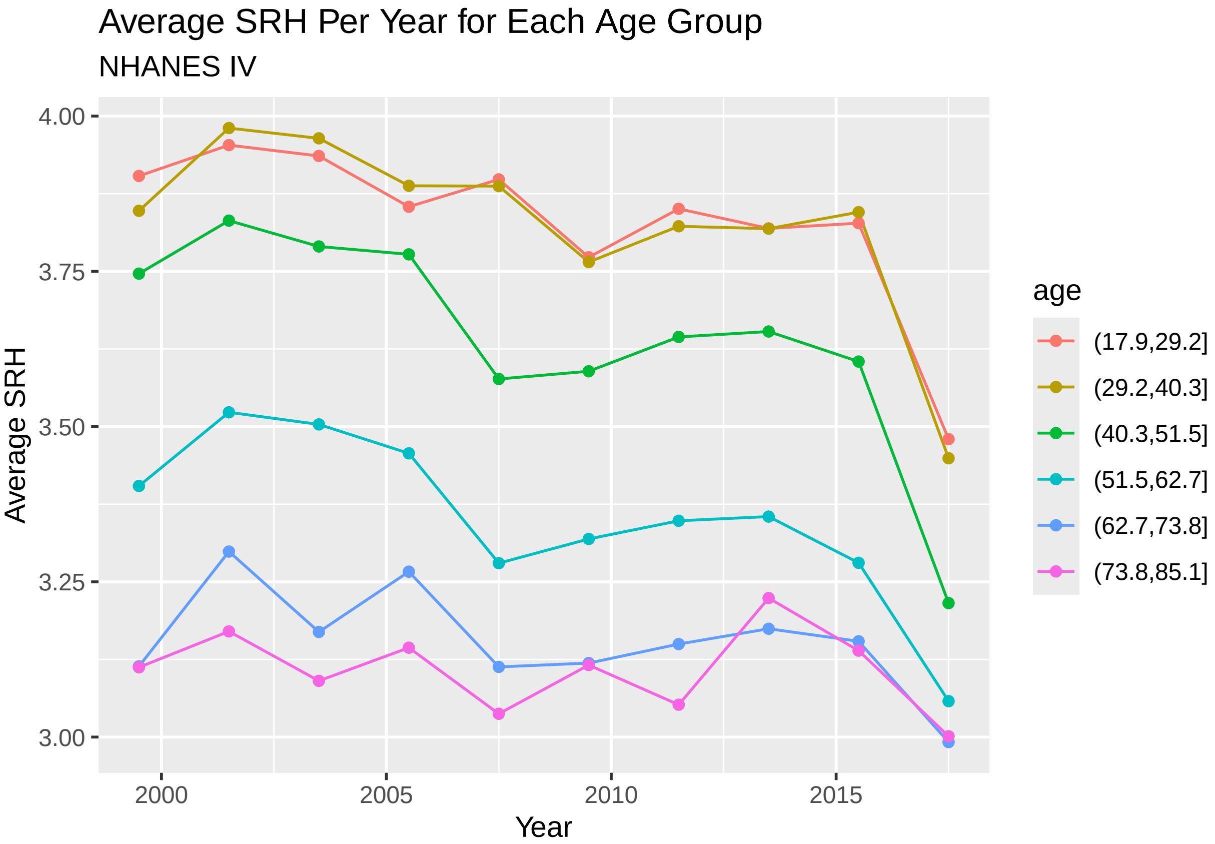
```
# Self-Rated Health Category Distribution by Age Group
data_nhanes %>%
  mutate(age_group = cut(age, breaks = 6)) %>% # Create cohorts with 6 breaks
  mutate(health = srh) %>%
  select(year, health, age_group) %>%
  filter(!is.na(health)) %>%
  mutate(
    srh = factor(health,
                 levels = 1:5,
                 labels = c("Poor", "Fair", "Good", "Very Good", "Excellent"))) %>%
  # Remove missing values
  # Calculate percentages by year
  group_by(year, age_group) %>%
  count(srh) %>%
  mutate(percent = n / sum(n) * 100) %>%
  ungroup() %>%
  ggplot(aes(x = year, y = percent, color = srh)) +
  geom_line() +
  geom_point() +
  labs(title = "Self-Rated Health Category Distribution",
       subtitle = "GSS Dataset",
       y = "Percent",
       x = "Year") +
  facet_wrap(~age_group) +
  theme_minimal()
```



Maybe the different NHANES studies aren't comparable – let's try only NHANES 4

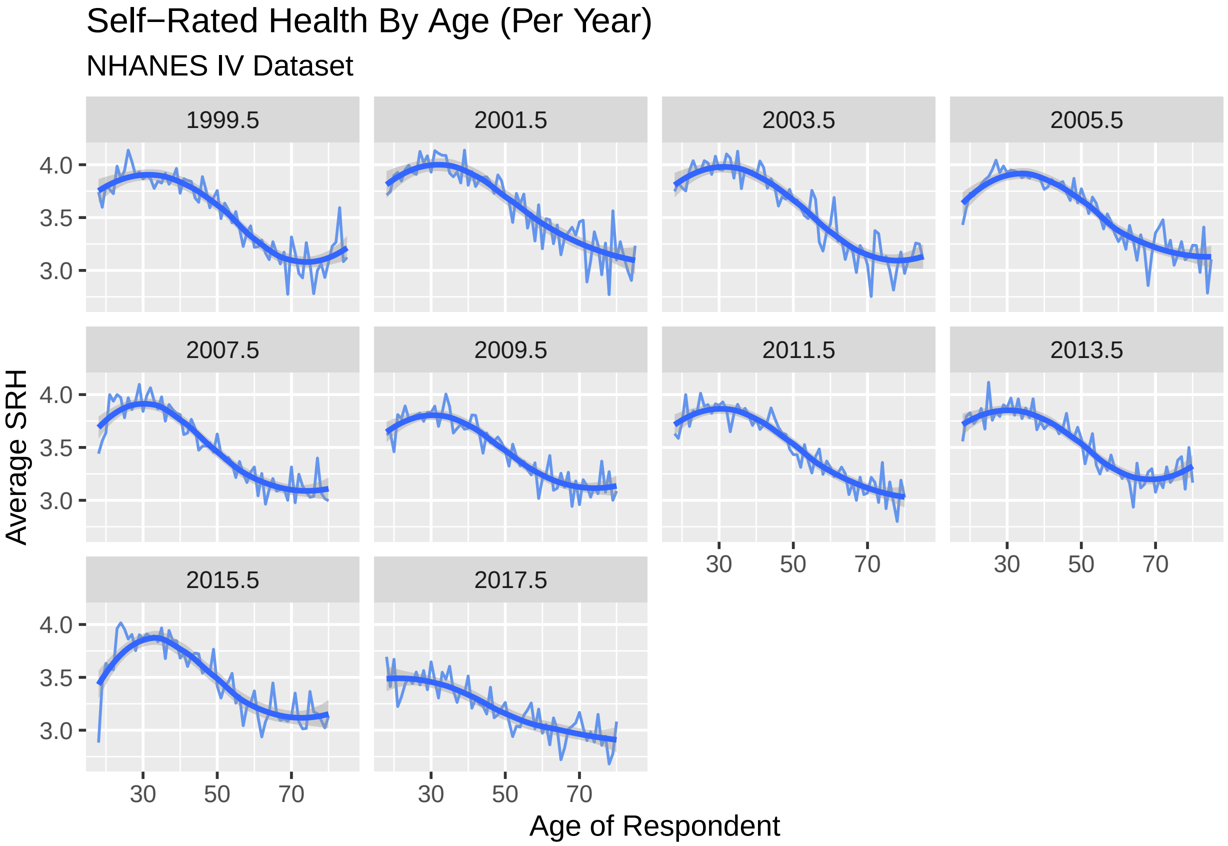
NHANES IV Only

```
nhanes4 <- nhanes4 %>%
  filter(age >= 18) %>%
  na.omit()

nhanes4 %>%
    mutate(age = cut(age, breaks = 6)) %>% # Create cohorts with 6 breaks
    group_by(age, year) %>%
    dplyr::summarize(mean_health = mean(srh, na.rm = TRUE), .groups = "drop") %>%
    ggplot(aes(x = year, y = mean_health, color = age)) +
    geom_line() +
    geom_point() +
    labs(title = "Average SRH Per Year for Each Age Group",
         subtitle = "NHANES IV",
         x = "Year",
         y = "Average SRH")
```



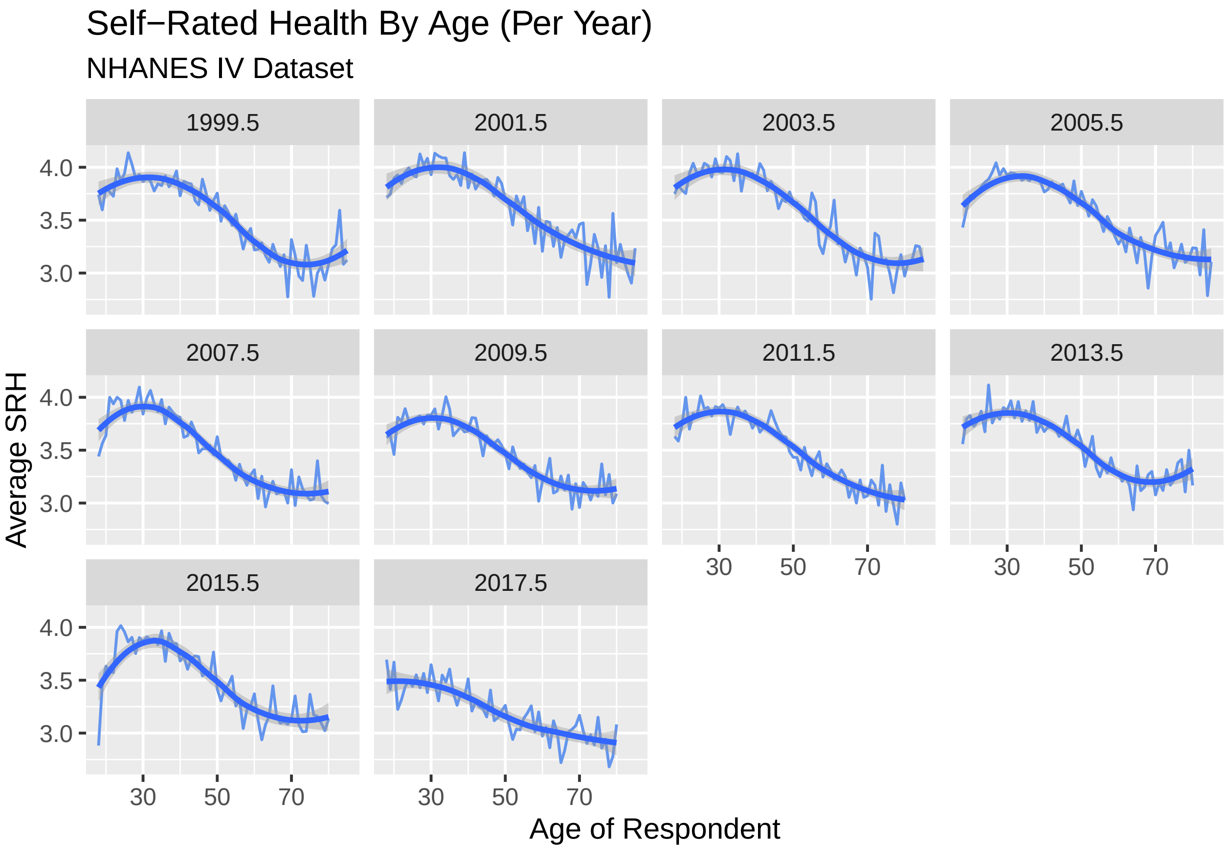
```
nhanes4 %>%
  group_by(age, year) %>%
  summarize(mean_health = mean(srh)) %>%
  ggplot(aes(x = age, y = mean_health)) +
  geom_line(color = "cornflowerblue") +
  geom_smooth() +
  facet_wrap(~ year) +
  labs(title = "Self-Rated Health By Age (Per Year)",
       subtitle = "NHANES IV Dataset",
       x = "Age of Respondent",
       y = "Average SRH",
       )
```

```
## `summarise()` has grouped output by 'age'. You can override using the `.groups`
## argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



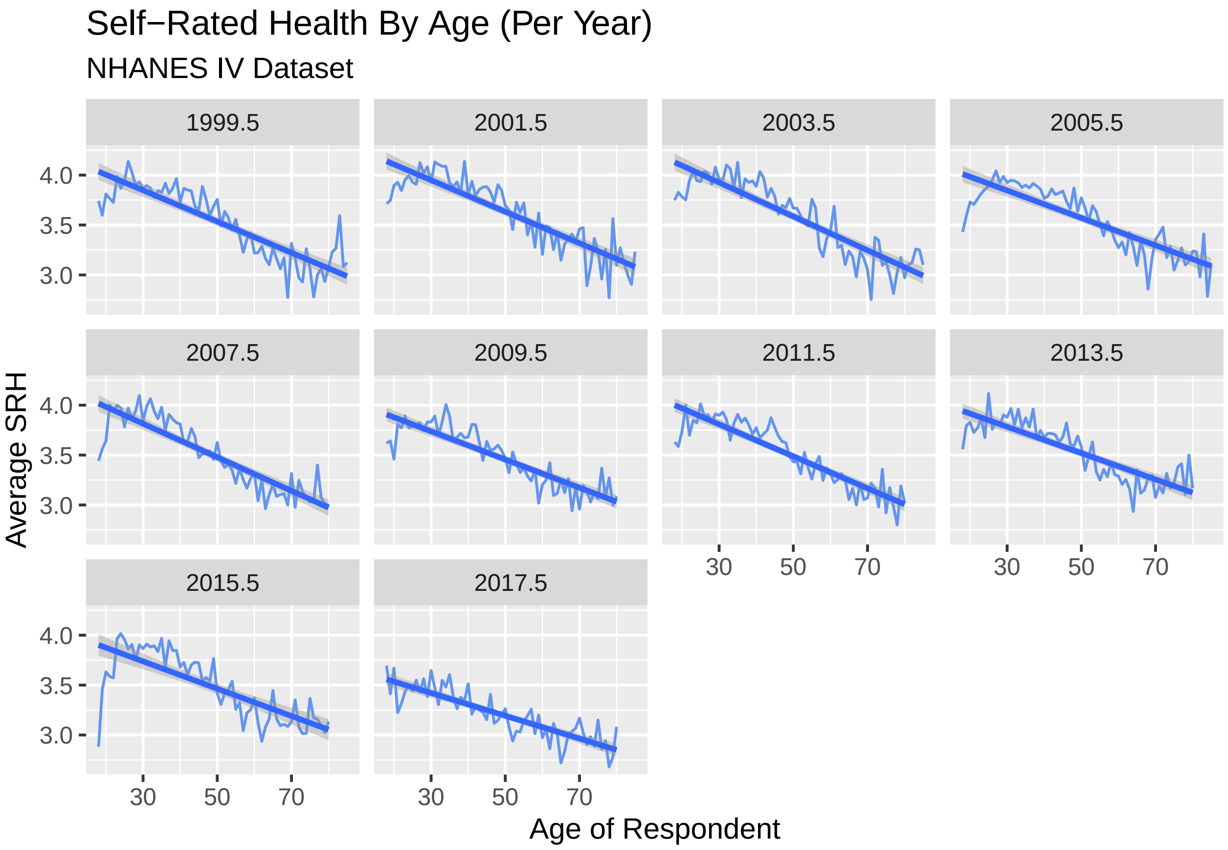
```
nhanes4 %>%
  group_by(age, year) %>%
  summarize(mean_health = mean(srh)) %>%
  ggplot(aes(x = age, y = mean_health)) +
  geom_line(color = "cornflowerblue") +
  geom_smooth() +
  facet_wrap(~ year) +
  labs(title = "Self-Rated Health By Age (Per Year)",
       subtitle = "NHANES IV Dataset",
       x = "Age of Respondent",
       y = "Average SRH",
       )
```

```
## `summarise()` has grouped output by 'age'. You can override using the `.groups`
## argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
nhanes4 %>%
  group_by(age, year) %>%
  summarize(mean_health = mean(srh)) %>%
  ggplot(aes(x = age, y = mean_health)) +
  geom_line(color = "cornflowerblue") +
  geom_smooth(method = "lm") +
  facet_wrap(~ year) +
  labs(title = "Self-Rated Health By Age (Per Year)",
       subtitle = "NHANES IV Dataset",
       x = "Age of Respondent",
       y = "Average SRH",
       )
```

```
## `summarise()` has grouped output by 'age'. You can override using the `.groups`
## argument.
## `geom_smooth()` using formula = 'y ~ x'
```



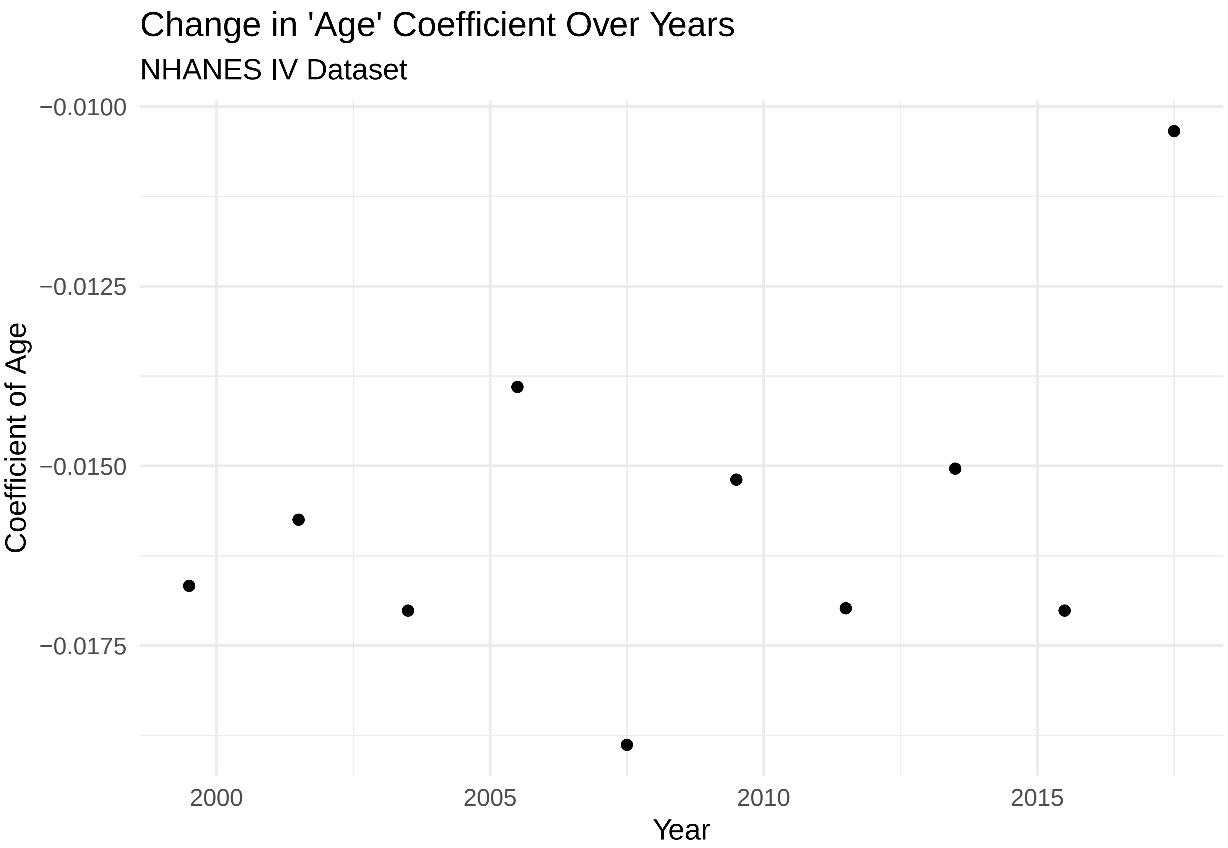
```r
library(broom)

# Aggregate slopes

# Perform linear regression for each year and extract the coefficient of 'age' with confidence interval
lm_health_v_age_0 <- nhanes4 %>%
  group_by(year) %>%
  do(tidy(lm(srh ~ age, data = .), conf.int = TRUE)) %>%  # Add conf.int = TRUE for CIs
  filter(term == "age") %>%
  select(year, coef = estimate, conf.low, conf.high, se = std.error, t_statistic = statistic,  p_value

# View the results with confidence intervals, se, t statistic, and p value
# print(lm_health_v_age_0)
knitr::kable(lm_health_v_age_0,
             caption = "NHANES IV")
```

Table 2: NHANES IV

year	coef	conf.low	conf.high	se	t_statistic	p_value
1999.5	-0.0166679	-0.0179948	-0.0153411	0.0006769	-24.62424	0
2001.5	-0.0157482	-0.0169775	-0.0145189	0.0006271	-25.11140	0
2003.5	-0.0170125	-0.0182480	-0.0157771	0.0006303	-26.99218	0
2005.5	-0.0139011	-0.0151836	-0.0126185	0.0006543	-21.24532	0
2007.5	-0.0188795	-0.0201763	-0.0175826	0.0006616	-28.53685	0
2009.5	-0.0151905	-0.0164678	-0.0139132	0.0006516	-23.31204	0
2011.5	-0.0169800	-0.0183033	-0.0156567	0.0006751	-25.15261	0
2013.5	-0.0150377	-0.0163551	-0.0137203	0.0006721	-22.37523	0

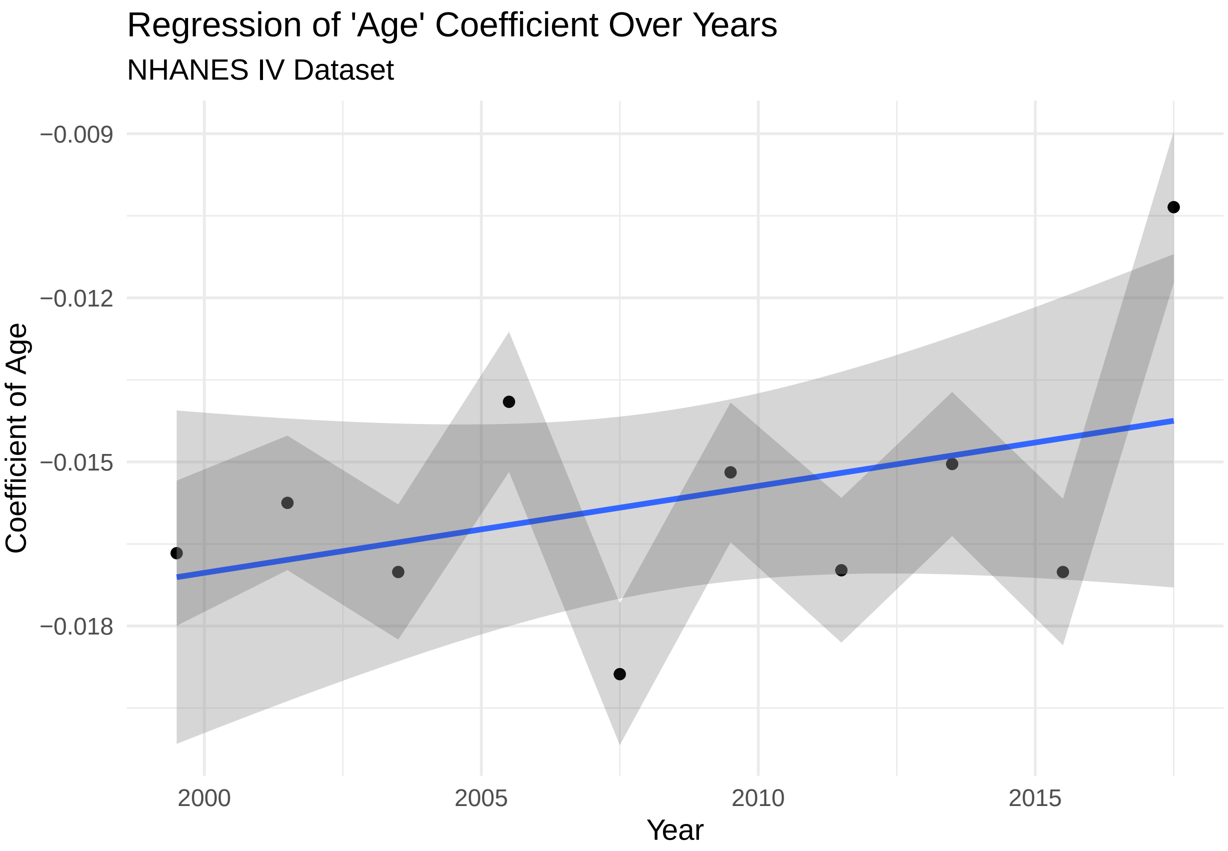
year	coef	conf.low	conf.high	se	t_statistic	p_value
2015.5	-0.0170124	-0.0183518	-0.0156729	0.0006833	-24.89712	0
2017.5	-0.0103421	-0.0117272	-0.0089570	0.0007066	-14.63698	0

```r
# Plot coefficients
ggplot(lm_health_v_age_0, aes(x = year, y = coef)) +
  geom_point() +
  labs(
    title = "Change in 'Age' Coefficient Over Years",
    subtitle = "NHANES IV Dataset",
    x = "Year",
    y = "Coefficient of Age"
  ) +
  theme_minimal()
```



```r
## Regress the srh vs age coefficients from each year on the year of the survey


# Visualize
ggplot(lm_health_v_age_0, aes(x = year, y = coef)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE) +  # Adds the regression line with standard error shading
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.2) +  # Confidence intervals for the co
  labs(
    title = "Regression of 'Age' Coefficient Over Years",
    subtitle = "NHANES IV Dataset",
    x = "Year",
    y = "Coefficient of Age"
  ) +
  theme_minimal()
```

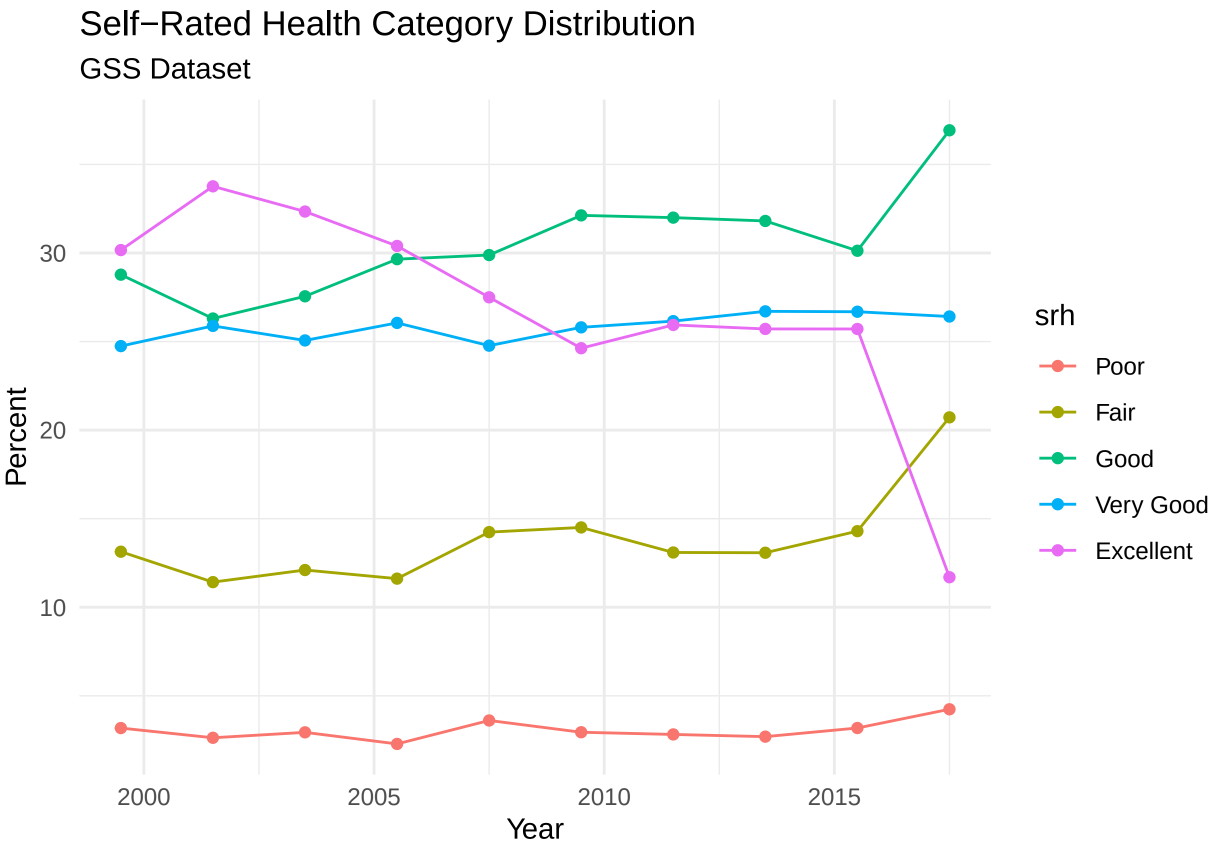
```
## `geom_smooth()` using formula = 'y ~ x'
```



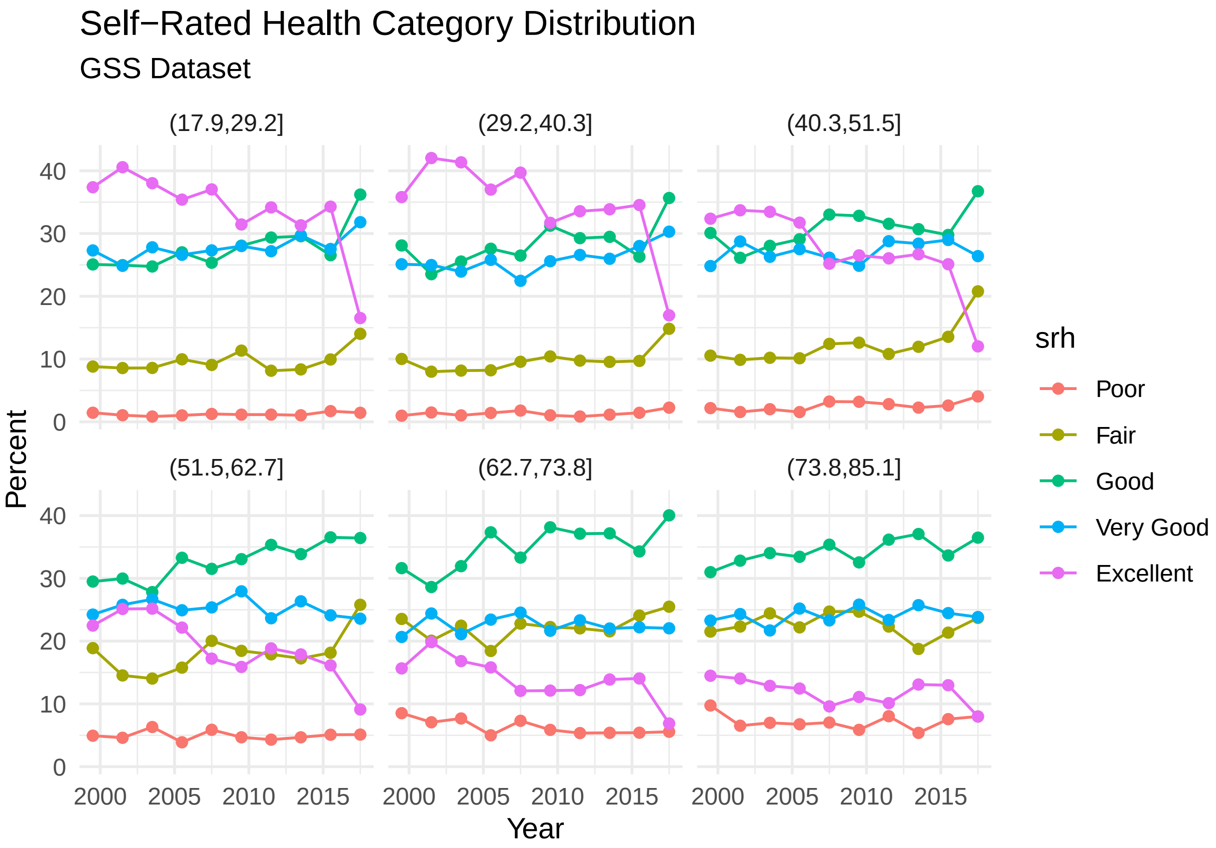
```
# Perform linear regression of 'coef' (age coefficient) vs 'year'
lm_coef_vs_year <- lm(coef ~ year, data = lm_health_v_age_0)

# View the summary of the regression
summary(lm_coef_vs_year)
```

```
##
## Call:
## lm(formula = coef ~ year, data = lm_health_v_age_0)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0030434 -0.0014698  0.0000866  0.0008902  0.0039057
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3346611  0.2485994  -1.346    0.215
## year         0.0001588  0.0001238   1.283    0.235
##
## Residual standard error: 0.002248 on 8 degrees of freedom
## Multiple R-squared:  0.1707, Adjusted R-squared:  0.06701
## F-statistic: 1.646 on 1 and 8 DF,  p-value: 0.2354
```

```
# Self-Rated Health Category Distribution
nhanes4 %>%
  mutate(health = srh) %>%
  select(year, health) %>%
  filter(!is.na(health)) %>%
  mutate(
    srh = factor(health,
                levels = 1:5,
                labels = c("Poor", "Fair", "Good", "Very Good", "Excellent"))) %>%
  # Remove missing values
  # Calculate percentages by year
  group_by(year) %>%
  count(srh) %>%
  mutate(percent = n / sum(n) * 100) %>%
  ungroup() %>%
  ggplot(aes(x = year, y = percent, color = srh)) +
  geom_line() +
  geom_point() +
  labs(title = "Self-Rated Health Category Distribution",
       subtitle = "GSS Dataset",
       y = "Percent",
       x = "Year") +
  theme_minimal()
```



```
# Self-Rated Health Category Distribution by Age Group
nhanes4 %>%
  mutate(age_group = cut(age, breaks = 6)) %>% # Create cohorts with 6 breaks
  mutate(health = srh) %>%
  select(year, health, age_group) %>%
  filter(!is.na(health)) %>%
  mutate(
    srh = factor(health,
                    levels = 1:5,
                    labels = c("Poor", "Fair", "Good", "Very Good", "Excellent"))) %>%
  # Remove missing values
  # Calculate percentages by year
  group_by(year, age_group) %>%
  count(srh) %>%
  mutate(percent = n / sum(n) * 100) %>%
  ungroup() %>%
  ggplot(aes(x = year, y = percent, color = srh)) +
  geom_line() +
  geom_point() +
  labs(title = "Self-Rated Health Category Distribution",
       subtitle = "GSS Dataset",
       y = "Percent",
       x = "Year") +
  facet_wrap(~age_group) +
  theme_minimal()
```


Self−Rated Health Category Distribution
GSS Dataset
(17.9,29.2]
(29.2,40.3]
(40.3,51.5]
(51.5,62.7]
(62.7,73.8]
(73.8,85.1]
Percent
Year
srh
Poor
Fair
Good
Very Good
Excellent