Investigate Hours Worked by Income Group

2024-01-26

Introduction

Federal Reserve System Data has consistently shown that the share of total net worth in the United States is held by individuals in the 99th percentile of wealth, while wealth in the middle percentiles has decreased, even among the 90th to 99th percentile. However, I believe this change is likely an underestimate of the decline in net worth across income groups. While net worth has been decreasing, I also hypothesize that hours worked per year are simultaneously increasing for individuals below the 99th percentile. In this way, net worth held by Americans might be comparable to "shrinkflation". Not only is relative income decreasing - simultaneously, the work required to obtain that income might be increasing.

To investigate this, I've extracted data through IPUMS for the CPS series between 1980 and 2020. This is cross-sectional microdata survey which contains demographic information, along with variables concerning work hours and income. I will use this data to see how hours of work have changed by year across income groups.

This R markdown notebook details how I extracted and processed the data, data analysis I performed to examine CPS variables, and some initial examinations which explore the relationship between income status and hours worked by year and demographic group.

Data processing

```
library(data.table)
library(ggplot2)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(srvyr)
```

```
##
## Attaching package: 'srvyr'

## The following objects are masked from 'package:plyr':
##
##     mutate, rename, summarise, summarize

## The following object is masked from 'package:stats':
##
##     filter
```

```
cps <- fread("./data/cps/cps_hours_worked_1980_2020.csv")

# Hold onto bare minimum columns we need for the analysis:
cps <- cps[, .(YEAR, MONTH, SERIAL, PERNUM, STATEFIP, GQ, HHINCOME,
               INCTOT, INCWELFR, INCRENT, EMPSTAT, AGE, SEX, MARST, RACE,
               WKSWORK1, UHRSWORKLY, WKSUNEM1, ASECWT)]

setnames(cps, names(cps), c("year", "month", "hhid", "pid", "state_fip",
                            "group_quarters", "hh_income", "p_income",
                            "income_welfare", "income_rent", "employment_status",
                            "age", "sex", "marital_status", "race",
                            "employed_weeks", "hours_per_week",
                            "unemployed_weeks", "survey_weight"))

# Only hold onto observations living in households
cps <- cps[group_quarters == 1,]

####################
# Record variables #
####################

# I created a state fips dataset to merge on state names using the codebook.
# I'm using the join function from the "plyr" package because I prefer the
# syntax better than the "merge" function in base R.

state_fips <- fread("./data/helper/state_fips.csv")
cps <- join(cps, state_fips, by = "state_fip", type = "left")

# Checking to see if there's any oddities in the states
unique(cps$state)
```

```
##  [1] "Maine"                "New Hampshire"        "Vermont"
##  [4] "Massachusetts"        "Rhode Island"         "Connecticut"
##  [7] "New York"             "New Jersey"           "Pennsylvania"
## [10] "Ohio"                 "Indiana"              "Illinois"
```

```
## [13] "Michigan"             "Wisconsin"            "Minnesota"
## [16] "Iowa"                 "Missouri"             "North Dakota"
## [19] "South Dakota"         "Nebraska"             "Kansas"
## [22] "Delaware"             "Maryland"             "District of Columbia"
## [25] "Virginia"             "West Virginia"        "North Carolina"
## [28] "South Carolina"       "Georgia"              "Florida"
## [31] "Kentucky"             "Tennessee"            "Alabama"
## [34] "Mississippi"          "Arkansas"             "Louisiana"
## [37] "Oklahoma"             "Texas"                "Montana"
## [40] "Idaho"                "Wyoming"              "Colorado"
## [43] "New Mexico"           "Arizona"              "Utah"
## [46] "Nevada"               "Washington"           "Oregon"
## [49] "California"           "Alaska"               "Hawaii"
```

```
#Nope

# Create new hhid, based on survey year. IPUMS CPS either provides a
# id based on longitudinal link or by year so we need a unique id
# across the series.
cps[, hhid := sprintf("%s_%s", year, hhid)]

# Remove children from the sample
cps <- cps[age >= 18,]

# Set ages above 90 to 90 since CPS started top-coding age in 1988:
cps[age >= 90, age := 90]

cps[, sex := ifelse(sex == 1, "male", "female")]

# Top-code race so that all the multiple categories are a single value. I'm
# doing this to make the recode easier:
cps[race > 800, race := 800]
cps[race >= 650 & race <= 652, race := 600]

new_race <- c("white", "black", "indigenous", "asian", "other", "multiple")

# Using the 'mapvalue' function from the plyr package to recode. I double-checked
# the coded values to ensure the order matched the new recodes.
cps[, race := mapvalues(race, sort(unique(cps$race)), new_race)]

# Coding both divorced and widowed as "former". Also, coding married with spouse
# as former, if hh_income is the same as personal income or married, if otherwise.
cps[marital_status %in% c(3, 4, 5, 7), marital_status := 3]

cps[marital_status == 2 & hh_income == p_income, marital_status := 3]
cps[marital_status == 2 & hh_income != p_income, marital_status := 1]

new_marry <- c("married", "former", "single")
cps[, marital_status := mapvalues(marital_status,
                                  sort(unique(cps$marital_status)),
                                  new_marry)]

# Simply recode employment status to either working, unemployed, or retired. Ignore
# the distinction between not in labor force and unemployed:
```

```r
cps[employment_status <= 12, employment_status := 10]
cps[employment_status >= 20 & employment_status <= 35, employment_status := 20]
cps[employment_status == 36, employment_status := 30]

new_employ <- c("employed", "unemployed", "retired")
cps[, employment_status := mapvalues(employment_status,
                                     sort(unique(cps$employment_status)),
                                     new_employ)]

cps[hours_per_week == 999, hours_per_week := NA]

# Check if there are any inconsistencies in number of weeks worked and
# employment status:
cps[employment_status != "employed" & employed_weeks > 0]
```

```
##         year month       hhid pid state_fip group_quarters hh_income p_income
##      1: 1980     3     1980_6   1        23              1      7798     7798
##      2: 1980     3     1980_8   3        23              1     17413     7760
##      3: 1980     3    1980_13   1        23              1     13700    13284
##      4: 1980     3    1980_16   3        23              1     34200     1909
##      5: 1980     3    1980_29   3        23              1      8097      557
##     ---
## 433367: 2023     3 2023_88935   2        15              1    225855   100041
## 433368: 2023     3 2023_88936   3        15              1     83500     3500
## 433369: 2023     3 2023_88951   1        15              1    105313    23752
## 433370: 2023     3 2023_88961   4        15              1    198401    58001
## 433371: 2023     3 2023_88965   4        15              1    166142     7200
##         income_welfare income_rent employment_status age    sex marital_status
##      1:              0          NA        unemployed  21   male         single
##      2:              0          NA        unemployed  24   male         former
##      3:              0          NA        unemployed  60   male        married
##      4:              0          NA        unemployed  18   male         single
##      5:              0          NA        unemployed  18   male         single
##     ---
## 433367:              0           0           retired  70   male        married
## 433368:              0           0        unemployed  19   male         single
## 433369:              0           0        unemployed  49 female        married
## 433370:              0           1        unemployed  45 female         former
## 433371:              0           0        unemployed  20 female         single
##             race employed_weeks hours_per_week unemployed_weeks survey_weight
##      1:    white             52             40               99        474.70
##      2:    white             30             40               22        447.86
##      3:    white             38             70               14        432.84
##      4:    white             10             40                0        460.40
##      5:    white              6             32                0        461.51
##     ---
## 433367:    asian             52             40               99        490.76
## 433368:    asian              8             20                0        402.17
## 433369: multiple              6             40                0        460.45
## 433370:    white             52             40               99        479.54
## 433371:    asian             24             30                0        757.88
##          state
##      1:  Maine
```

```
##      2:  Maine
##      3:  Maine
##      4:  Maine
##      5:  Maine
##     ---
## 433367: Hawaii
## 433368: Hawaii
## 433369: Hawaii
## 433370: Hawaii
## 433371: Hawaii
```

```
# Looking at codebook, this variable indicates if they were unemployed within
# 4 weeks of when the survey was issued, rather than if they were unemployed
# for a given period of time. So I'll need to keep this in mind.

cps[unemployed_weeks == 99, unemployed_weeks := NA]

# Check to see if there are any inconsistencies between unemployed weeks and
# weeks worked:
cps[, total_weeks := employed_weeks + unemployed_weeks]

sort(unique(cps$total_weeks))
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## [51] 51 52 55 56 57 58 60 61 62
```

```
# This looks strange - let's look at a cross section to see if we
# can figure out what's going on
cps[total_weeks == 4,]
```

```
##        year month       hhid pid state_fip group_quarters hh_income p_income
##     1: 1980     3   1980_110   2        23              1     15096      115
##     2: 1980     3   1980_818   4        23              1      5990     2640
##     3: 1980     3  1980_1028   2        23              1      6292     2476
##     4: 1980     3  1980_1147   1        23              1      5108     5108
##     5: 1980     3  1980_1691   2        33              1      7246      315
##    ---
## 13277: 2023     3 2023_86574   1         2              1     30391     9550
## 13278: 2023     3 2023_87220   3         2              1    366035    23284
## 13279: 2023     3 2023_87239   1         2              1    241582    74798
## 13280: 2023     3 2023_87849   4        15              1    118128    25603
## 13281: 2023     3 2023_88392   5        15              1     29000     2000
##        income_welfare income_rent employment_status age    sex marital_status
##     1:              0          NA        unemployed  53 female        married
##     2:           2340          NA          employed  23 female         former
##     3:           2370          NA        unemployed  45 female        married
##     4:           4508          NA          employed  42 female         former
##     5:              0          NA          employed  45 female        married
##    ---
## 13277:              0           1           retired  63 female         former
## 13278:              0           0        unemployed  22   male         former
## 13279:              0        6000          employed  73 female        married
```

```
## 13280:              0           0           retired  71 female         former
## 13281:              0           0        unemployed  21 female         single
##              race employed_weeks hours_per_week unemployed_weeks survey_weight
##     1:      white              4             16                0        485.08
##     2:      white              4             30                0        473.83
##     3:      white              4              8                0        437.16
##     4:      white              4             40                0        378.67
##     5:      white              4             30                0        492.64
##    ---
## 13277:      white              2             40                2        403.61
## 13278: indigenous              4             70                0        515.32
## 13279:      white              4             15                0        285.90
## 13280:      white              4              2                0        352.65
## 13281:      asian              4             40                0        789.92
##                state total_weeks
##     1:         Maine           4
##     2:         Maine           4
##     3:         Maine           4
##     4:         Maine           4
##     5: New Hampshire           4
##    ---
## 13277:        Alaska           4
## 13278:        Alaska           4
## 13279:        Alaska           4
## 13280:        Hawaii           4
## 13281:        Hawaii           4
```

```
# For values less than 52, I'm guessing this comes down to unemployment being
# a very specific definition. What about more than 52 weeks?
cps[total_weeks > 52]
```

```
##     year month       hhid pid state_fip group_quarters hh_income p_income
##  1: 2001     3 2001_20959   3        42              1     77285    17050
##  2: 2002     3 2002_43963   3        20              1     53446    14000
##  3: 2003     3  2003_2416   3        33              1    113503    51000
##  4: 2004     3 2004_13215   2        36              1     17500    17500
##  5: 2004     3 2004_14104   1        36              1     16896    16896
##  6: 2004     3 2004_21694   1        39              1     44140    14000
##  7: 2004     3 2004_47960   3        11              1     10424     2756
##  8: 2004     3 2004_90938   2         6              1     46885    23000
##  9: 2004     3 2004_95192   6         6              1     35617    16896
## 10: 2005     3 2005_77250   4        56              1     36121    21000
## 11: 2006     3 2006_27757   2        26              1     11418     4000
## 12: 2006     3 2006_80662   5         4              1     42348    10000
## 13: 2006     3 2006_95963   4        15              1    129313     4640
## 14: 2007     3 2007_13242   2        36              1     32450    18020
## 15: 2007     3 2007_21441   2        39              1     54425    18019
## 16: 2007     3 2007_31742   2        27              1     38969    28868
## 17: 2008     3 2008_63477   2         1              1     55320    19120
##     income_welfare income_rent employment_status age    sex marital_status
##  1:              0           0          employed  21   male         single
##  2:              0           0          employed  25   male         single
##  3:              0           0        unemployed  20   male         single
##  4:              0           0          employed  28   male        married
```

```
##  5:              0           0        unemployed  41 female         single
##  6:              0           0          employed  19   male         single
##  7:           1156           0          employed  19 female         single
##  8:              0           0        unemployed  23   male         single
##  9:              0           0          employed  39 female         single
## 10:              0           0        unemployed  46 female         former
## 11:              0           0        unemployed  27   male         single
## 12:              0           0          employed  26   male         single
## 13:              0           0          employed  19 female         single
## 14:              0           0          employed  22   male         single
## 15:              0           0          employed  22   male         single
## 16:              0           0        unemployed  50   male        married
## 17:              0           0          employed  43   male        married
##         race employed_weeks hours_per_week unemployed_weeks survey_weight
##  1:    white             32              38               26       4856.46
##  2:    white             48              45               12        966.59
##  3:    white             30              40               26        491.02
##  4:    white             16              40               40       1887.19
##  5:    black             46              40               12       1743.48
##  6:    white             35              40               20        958.39
##  7:    black             24              40               32        223.12
##  8:    white             25              20               32       5662.88
##  9:    black             46              40               12       1598.07
## 10:    white             35              35               22        131.57
## 11:    black             20              40               35       3380.37
## 12:    asian             20              40               40       1519.24
## 13: multiple             30              20               26        338.58
## 14:    black             36              40               26       2075.11
## 15:    black             36              40               26       2740.25
## 16:    black             35              38               26        718.28
## 17:    white             24              30               32       2924.93
##                     state total_weeks
##  1:         Pennsylvania          58
##  2:               Kansas          60
##  3:        New Hampshire          56
##  4:             New York          56
##  5:             New York          58
##  6:                 Ohio          55
##  7: District of Columbia          56
##  8:           California          57
##  9:           California          58
## 10:              Wyoming          57
## 11:             Michigan          55
## 12:              Arizona          60
## 13:               Hawaii          56
## 14:             New York          62
## 15:                 Ohio          62
## 16:            Minnesota          61
## 17:              Alabama          56
```

```
# There's only 17 observations where this occurs so I'm guessing there's a
# miscoding problem. I'll impute unemployed weeks for these 17 individuals as
# being the difference between 52 weeks and weeks worked:
cps[total_weeks > 52, unemployed_weeks := 52 - employed_weeks]
```

```r
# I had to look at the IPUMS CPS on top-coding to get a better sense on which
# variables I should modify concerning income:
# https://cps.ipums.org/cps/topcodes_tables.shtml

# However, the histograms of both hh_income and p_income appear to not be top-coded
# Not sure what's going on here (extends all the way to 3 million!)
quantile(cps$hh_income, probs = seq(0, 1, 0.1))
```

```
##      0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
##  -37040   11403   19396   27059   35511   45286   57286   73000   95484  135800
##    100%
## 3300477
```

```r
# Convert all income into 2022 dollars. I used the suggested inflation factors
# from IPUMS: https://cps.ipums.org/cps/cpi99.shtml
inflate <- fread("./data/helper/in_2022_dollars.csv")
cps <- join(cps, inflate, by = "year", type = "left")

cps[, hh_income := hh_income*adjustment]
cps[, p_income := p_income*adjustment]

# Let's create a new variable which corresponds to the FRED categories for income
# share. We'll need to calculate this separately for each year:

cps[, hh_income_rank := percent_rank(.SD$hh_income), by = "year"]
cps[, p_income_rank := percent_rank(.SD$p_income), by = "year"]

# Alternatively, I could calculate this using survey weights.

hh_income_share <- function(df){

  hh_inc <- df$hh_income
  df <- as_survey_design(df, weights = survey_weight)

  # The idea here is to calculate weighted quantiles of household income, then
  # use these quantiles to create categorical groups. To do that, once I get
  # the quantiles, I will create a new dataframe which holds the upper value
  # for each quantile. Then, I'll use the "cut" command in base
  # R to map these quantiles to a new categorical group.

  breaks <- df %>%
            summarise(perc = survey_quantile(hh_income, c(0.01, 0.5, 0.9, 0.99, 1),
                                             vartype = NULL))

  breaks <- melt(as.data.table(breaks),
                 variable.name = "percentile", value.name = "income")

  df <- as.data.table(df$variables)

  income_groups <- c("[0 - 1)", "[1, 50)", "[50, 90)", "[90, 99)", "99+")

  # Note: For cut to work, I need breaks to include the lowest value in the data
  # as a starting value:
```

```
  percentile_categories <- cut(hh_inc, breaks = c(min(hh_inc), breaks$income),
                               labels = income_groups,
                               right = F, include.lowest = T)
  return(percentile_categories)

}

cps[, hh_income_group := hh_income_share(.SD), by = "year"]

# Let's also use the naive household income rank to create income percentile groups
# matching FRED, then see how well these compare to the weighted version.

income_groups <- c("[0 - 1)", "[1, 50)", "[50, 90)", "[90, 99)", "99+")
cps[, hh_income_group_unweighted := cut(hh_income_rank,
                                        breaks = c(0, 0.01, 0.5, 0.9, 0.99, 1),
                                        labels = income_groups,
                                        right = F, include.lowest = T)]

# How often does the unweighted estimate not match the weighted estimate? Get number
# of rows in the dataset, then compare to number of rows in filtered dataset
# where income groups do not match.
n_total <- dim(cps)[1]
n_diff  <- dim(cps[hh_income_group != hh_income_group_unweighted])[1]

n_diff/n_total
```

```
## [1] 0.0176116
```

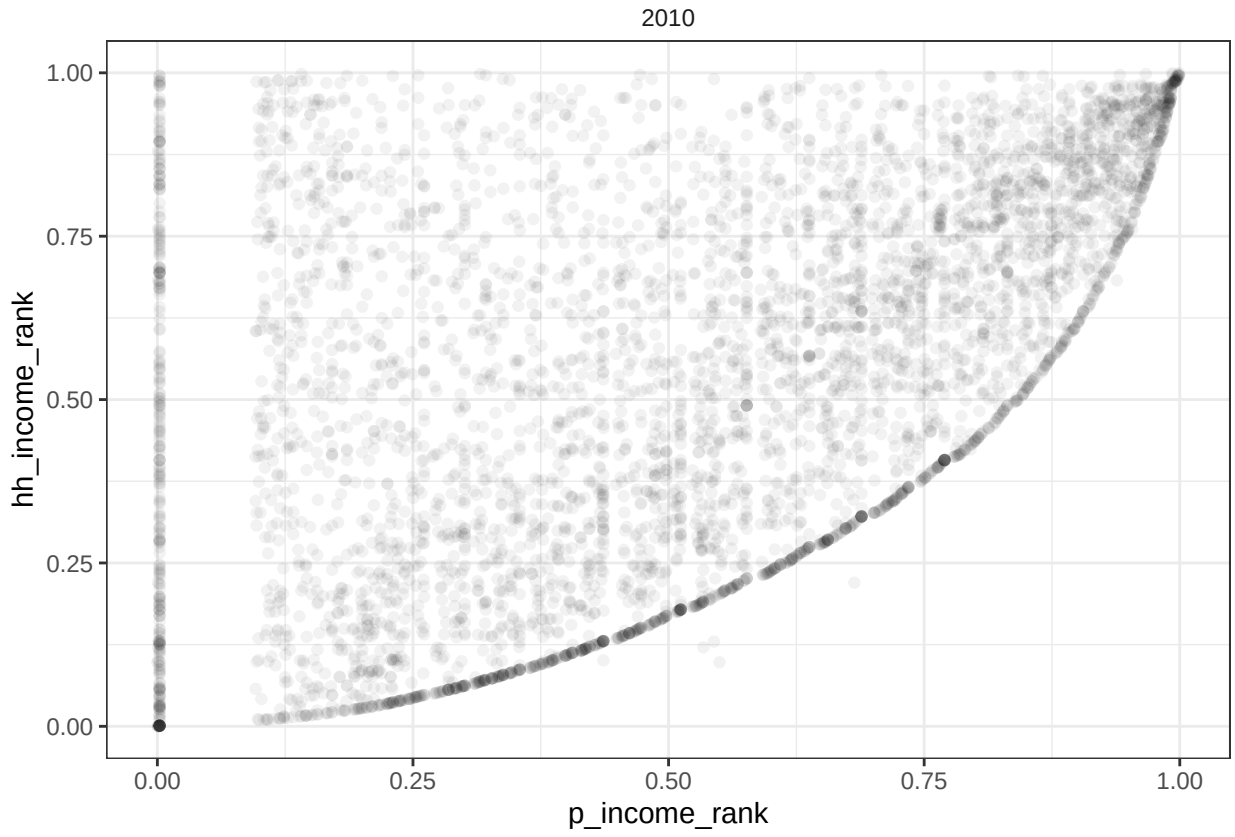
```
# Only 1.5% of the sample isn't matching. This doesn't seem like a huge deal
# to me so I'll neglect to incorporate weights for now.

# Income welfare and income rent have substantial missingness and a lack of variation
# So let's just drop these:

cps[, income_rent := NULL]
cps[, income_welfare := NULL]
cps[, total_weeks := NULL]
cps[, adjustment := NULL]
```

Data explorations

Let's start by looking at some basic patterns in the data to better understand if we need to make any transformations or outlier any data points. I'll start by looking at some probability densities by variable, then also look at some scatterplots. What I chose to do here was mostly exploratory, based off my interest in the data. The hope is to better interpret the data and find problems while doing this series of explorations.

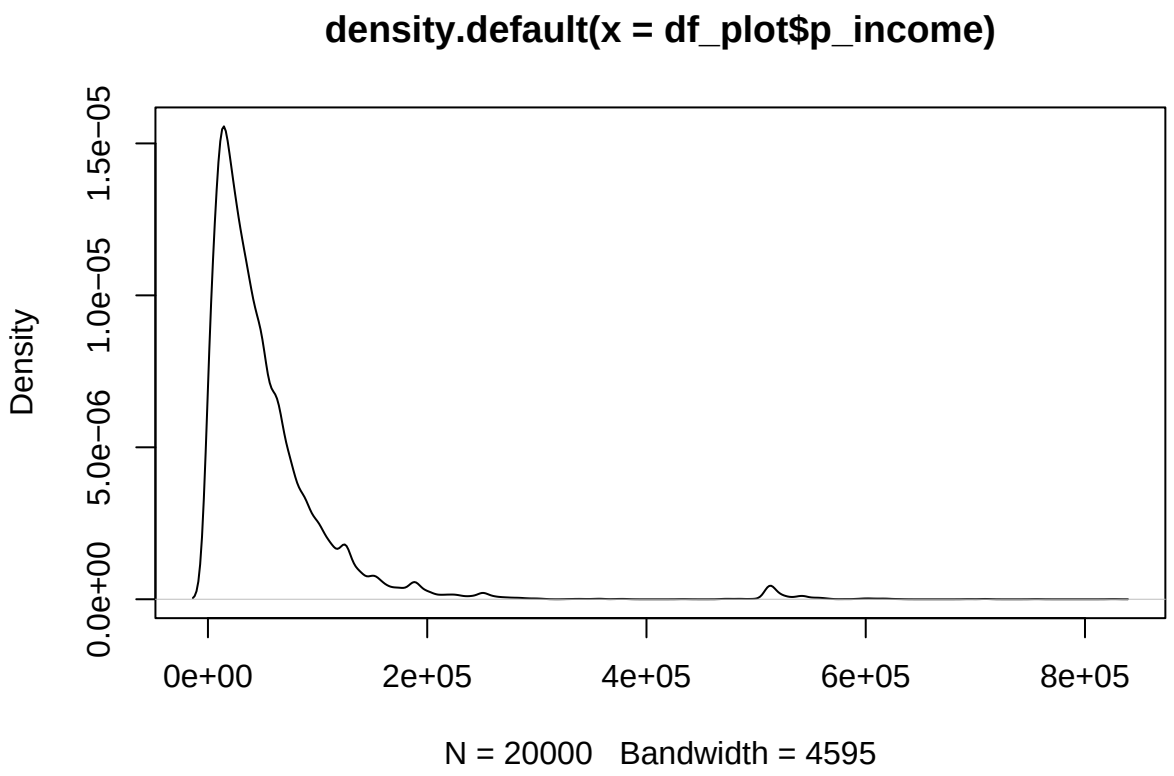
```
# How balanced is household income rank with personal income rank?

# I looked at the below plot by year and also by looking at the full sample.
# This subset captures the general pattern, which is similar across years
# and total sample.
```

```
df_plot <- cps[year == 2010, .SD[sample(5000)]]

ggplot(df_plot, aes(x = p_income_rank, y = hh_income_rank)) +
  geom_point(alpha = 0.05) +
  facet_wrap(~year) +
  theme_bw() +
  theme(strip.background = element_blank())
```

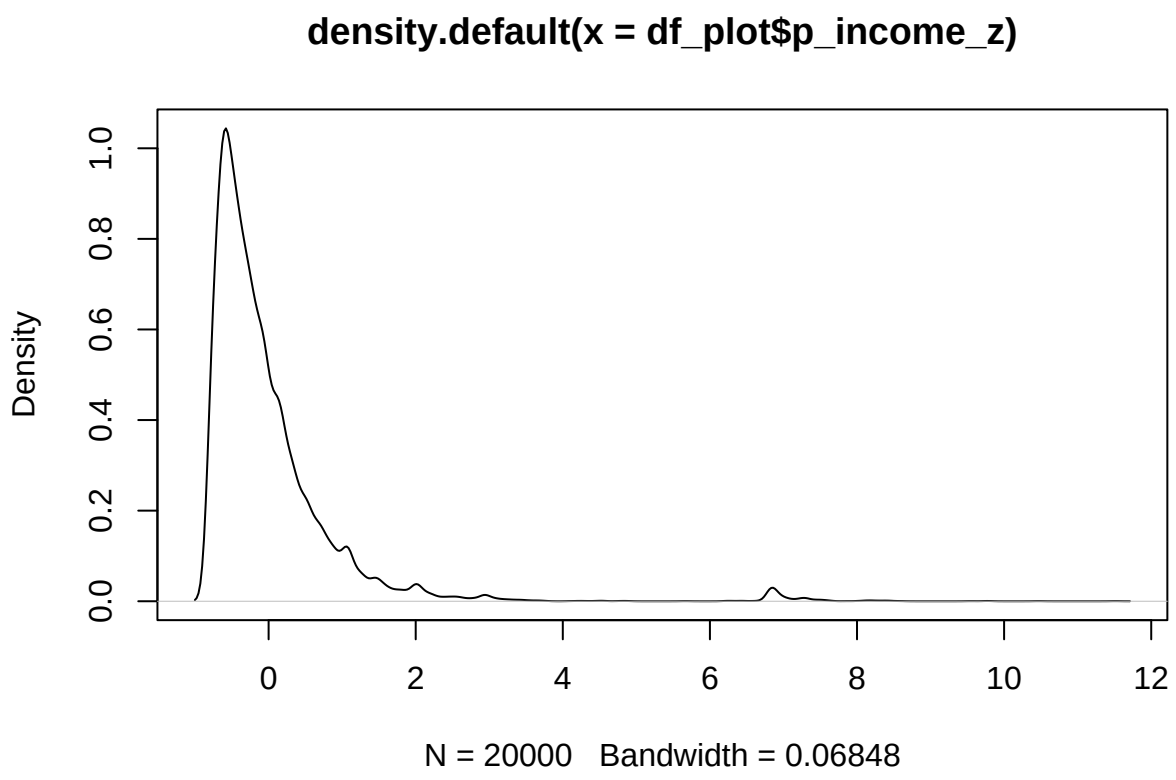


Really interesting plot. Large fraction of observations are likely in single-earner households (points at zero personal income). There's also a clear pattern in the income rank which makes sense - if your personal income is sufficiently large, then there's no way you would be below a given household income rank. There's also clustering in points along that line.

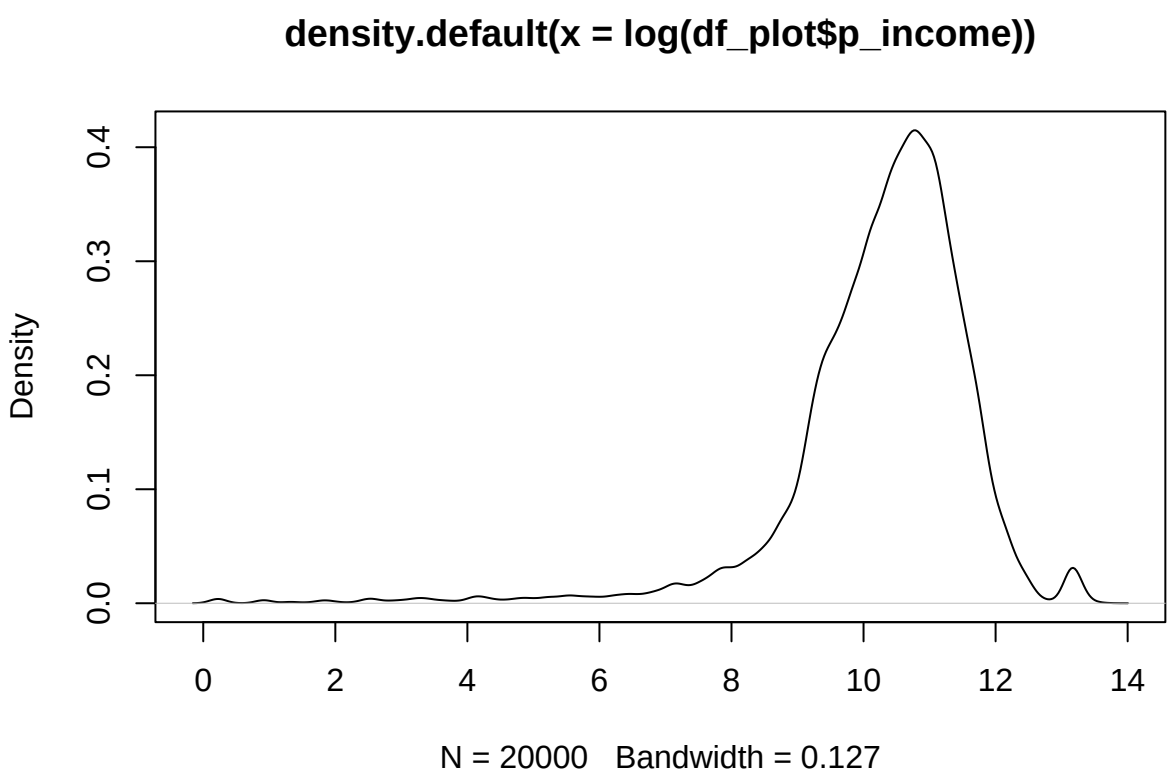
```
# This is didactic but look at distribution of personal income and see
# results of transforming this variable:

df_plot <- cps[year == 2010 & p_income > 0, .SD[sample(20000)]]
plot(density(df_plot$p_income))
```




```
# Same plot but using standard scores:
df_plot[, p_income_z := (p_income - mean(p_income))/sd(p_income)]
plot(density(df_plot$p_income_z))
```

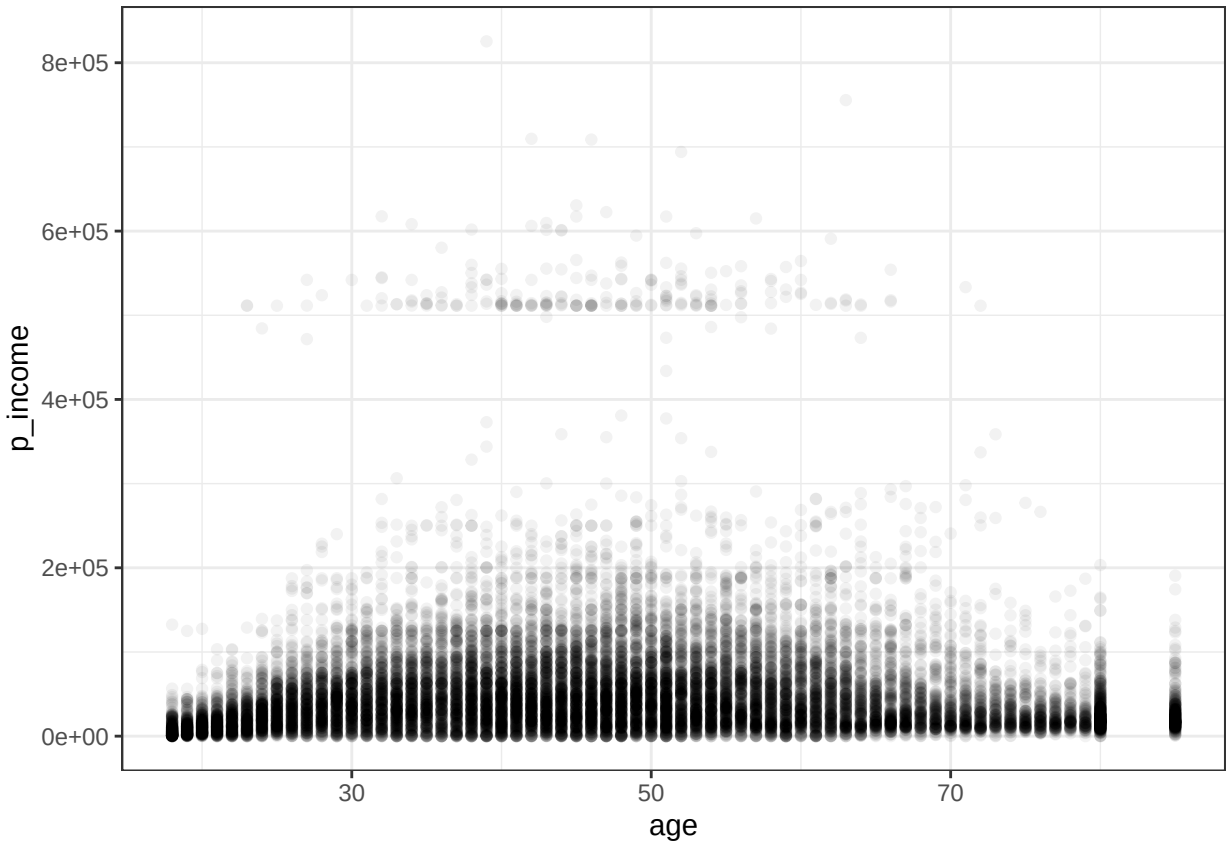



```
# Same shape! What if we log the variable?
plot(density(log(df_plot$p_income)))
```




```
# Shape is more symmetrical, though as discussed in the lecture this is not
# the important part for modeling!
```

```
ggplot(df_plot, aes(y = p_income, x = age)) +
  geom_point(alpha = 0.05) +
  theme_bw()
```

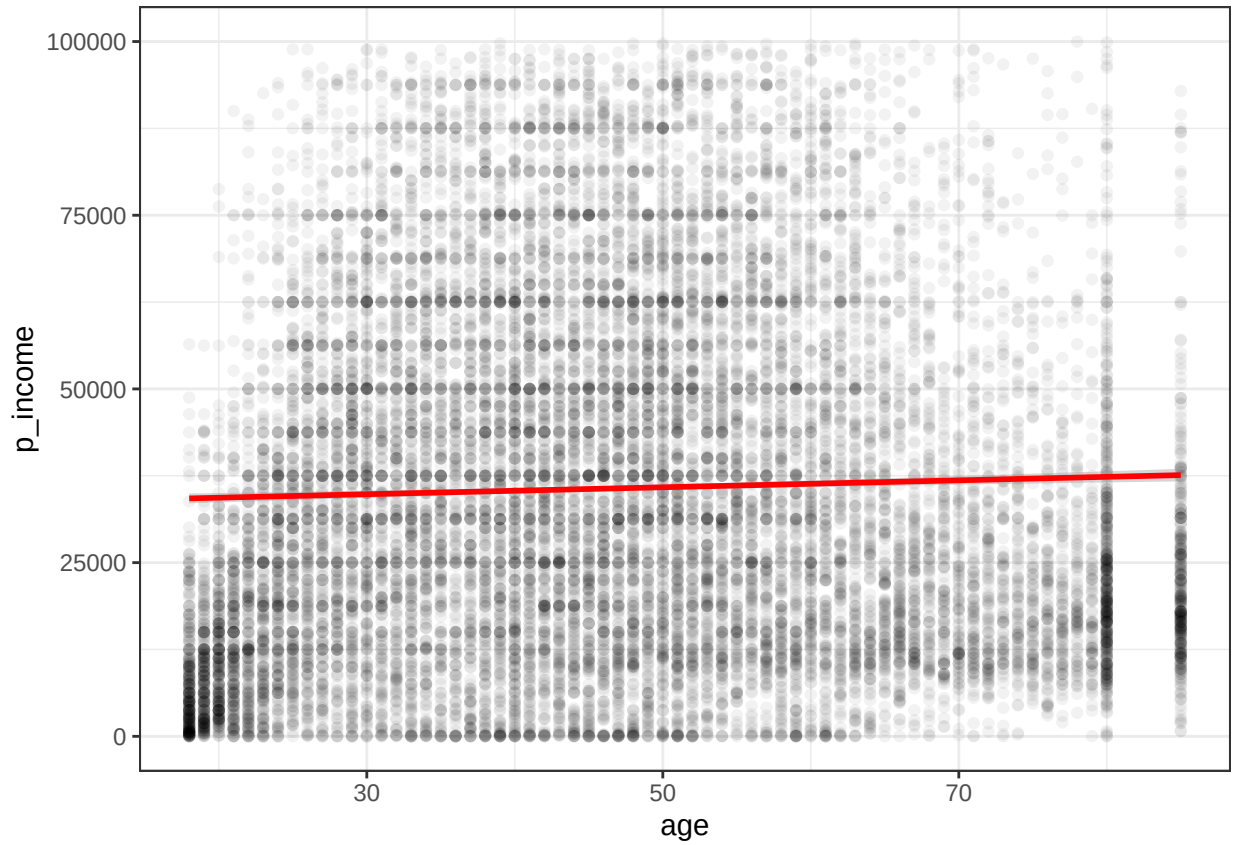



```
summary(lm(p_income ~ age, df_plot))
```

```
##
## Call:
## lm(formula = p_income ~ age, data = df_plot)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57557 -37024 -16901  13127 772866
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47764.67    1408.60  33.909  < 2e-16 ***
## age           122.42      28.33   4.322 1.55e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67060 on 19998 degrees of freedom
## Multiple R-squared:  0.0009332,  Adjusted R-squared:  0.0008832
## F-statistic: 18.68 on 1 and 19998 DF,  p-value: 1.554e-05
```

```
ggplot(df_plot, aes(y = p_income, x = age)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method='lm', formula = y ~ x, color = "red") +
```

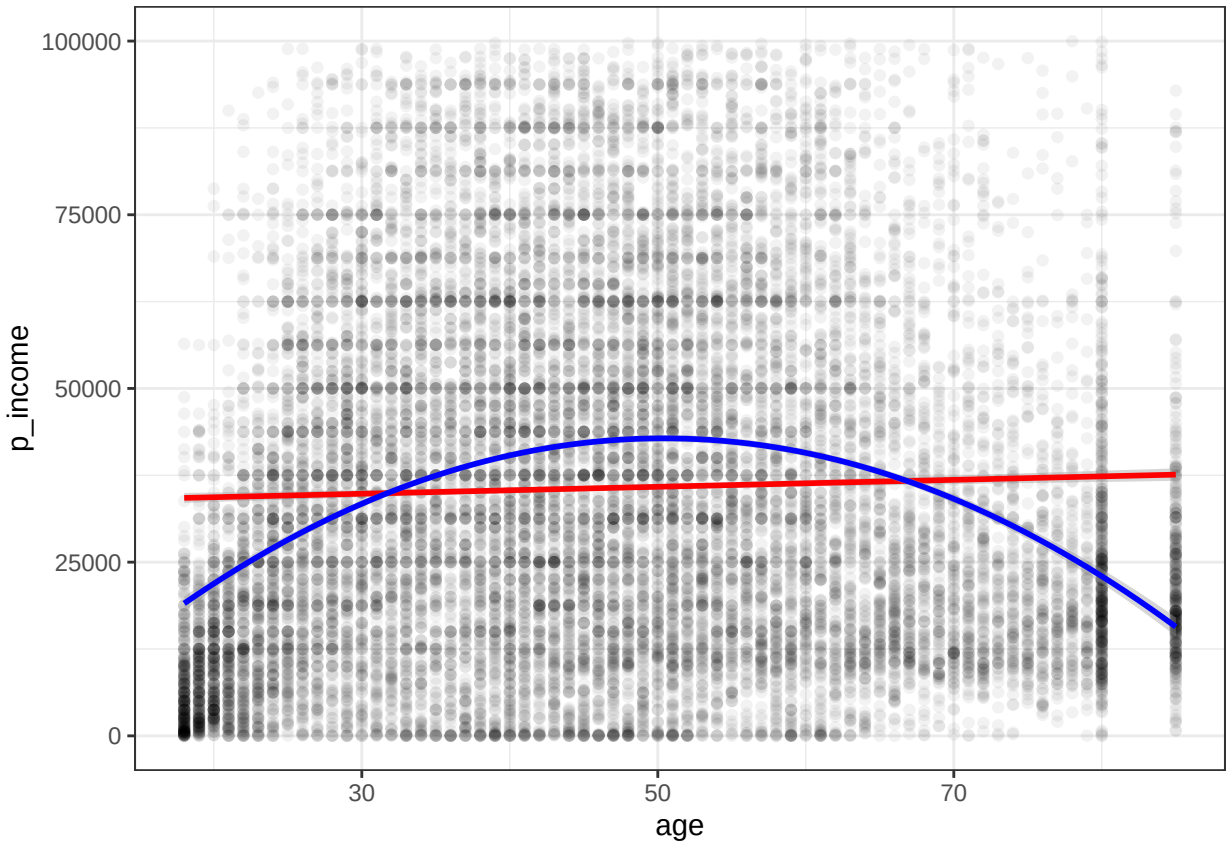
```
  lims(y = c(0, 1e5)) +
  theme_bw()
```




```
summary(lm(p_income ~ age + I(age^2), df_plot))
```

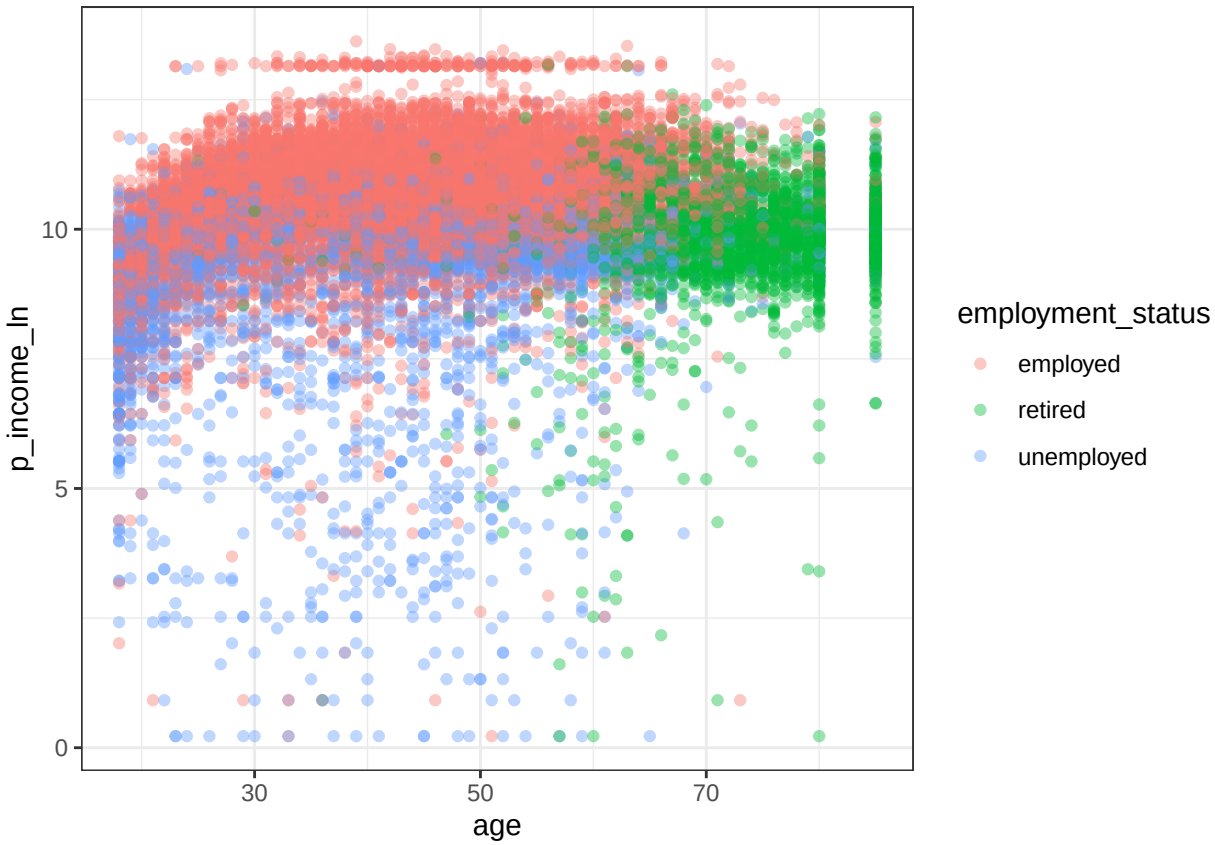
```
##
## Call:
## lm(formula = p_income ~ age + I(age^2), data = df_plot)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -68091 -32738 -12511  11855 763878
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -58714.67    3392.34  -17.31   <2e-16 ***
## age           5022.46     145.47   34.53   <2e-16 ***
## I(age^2)       -49.73       1.45  -34.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65170 on 19997 degrees of freedom
## Multiple R-squared:  0.05646,    Adjusted R-squared:  0.05637
## F-statistic: 598.3 on 2 and 19997 DF,  p-value: < 2.2e-16
```

```
ggplot(df_plot, aes(y = p_income, x = age)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method='lm', formula = y ~ x, color = "red") +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), color = "blue") +
  lims(y = c(0, 1e5)) +
  theme_bw()
```



Both models are under performing but there is some intuition behind the squared model. We can see in the figure that there's a ton of data points with smaller incomes in the high age categories. So maybe retirees are driving the negative effect on squared age, which seems implausible (most people have increasing personal income by age).

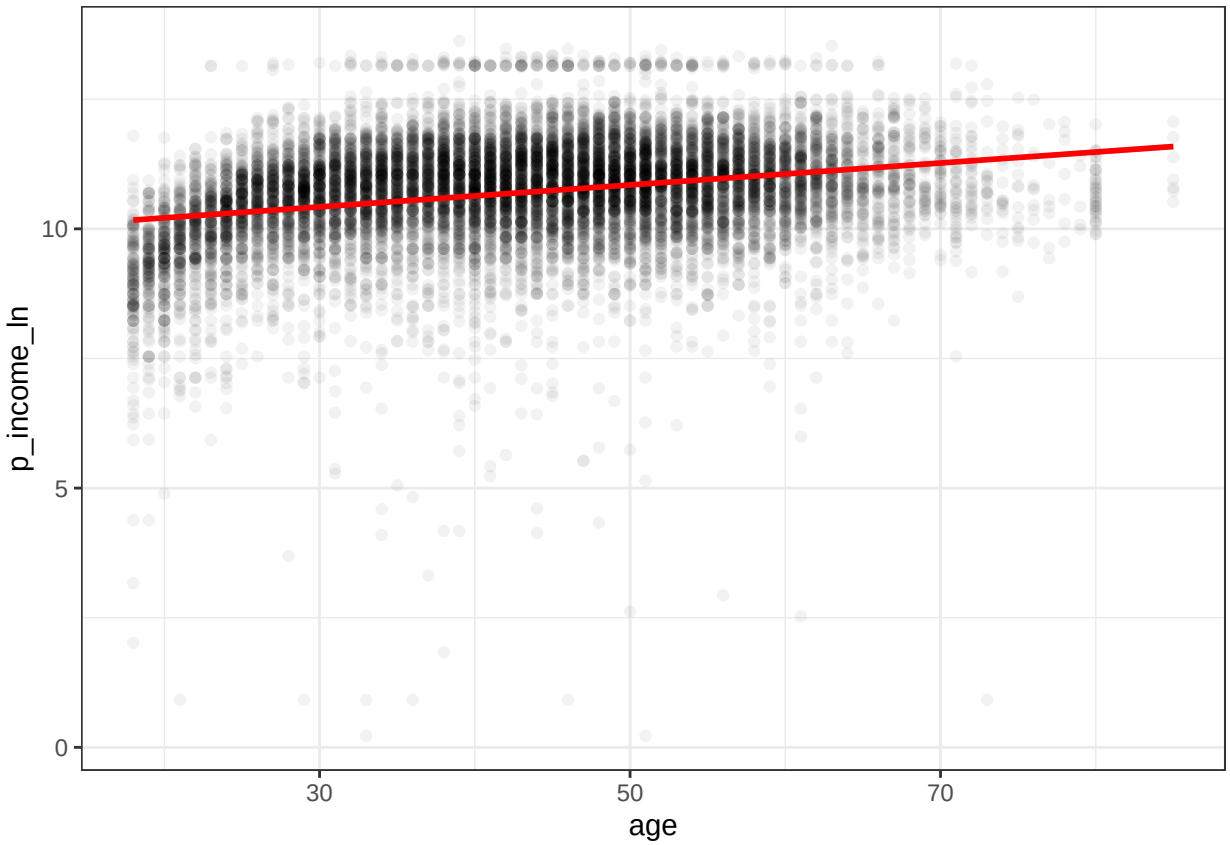
```
df_plot[, p_income_ln := log(p_income)]

ggplot(df_plot, aes(y = p_income_ln, x = age, color = employment_status)) +
  geom_point(alpha = 0.4) +
  theme_bw()
```




```
# A bit easier to see patterns now. Let's restrict sample to employed individuals
# and now use logged income as the dependent variable

df_plot_subset <- df_plot[employment_status == "employed",]

mod1 <- lm(p_income_ln ~ age, df_plot_subset)
summary(mod1)
```

```
##
## Call:
## lm(formula = p_income_ln ~ age, data = df_plot_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6447  -0.4758   0.0924   0.5896   3.0098
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.7883509  0.0291345  335.97   <2e-16 ***
## age         0.0211671  0.0006509   32.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.981 on 13511 degrees of freedom
## Multiple R-squared:  0.07259,    Adjusted R-squared:  0.07252
## F-statistic:  1057 on 1 and 13511 DF,  p-value: < 2.2e-16
```

```
df_plot_subset[, preds1 := predict(mod1)]

ggplot(df_plot_subset, aes(y = p_income_ln, x = age)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method='lm', formula = y ~ x, color = "red") +
  theme_bw()
```

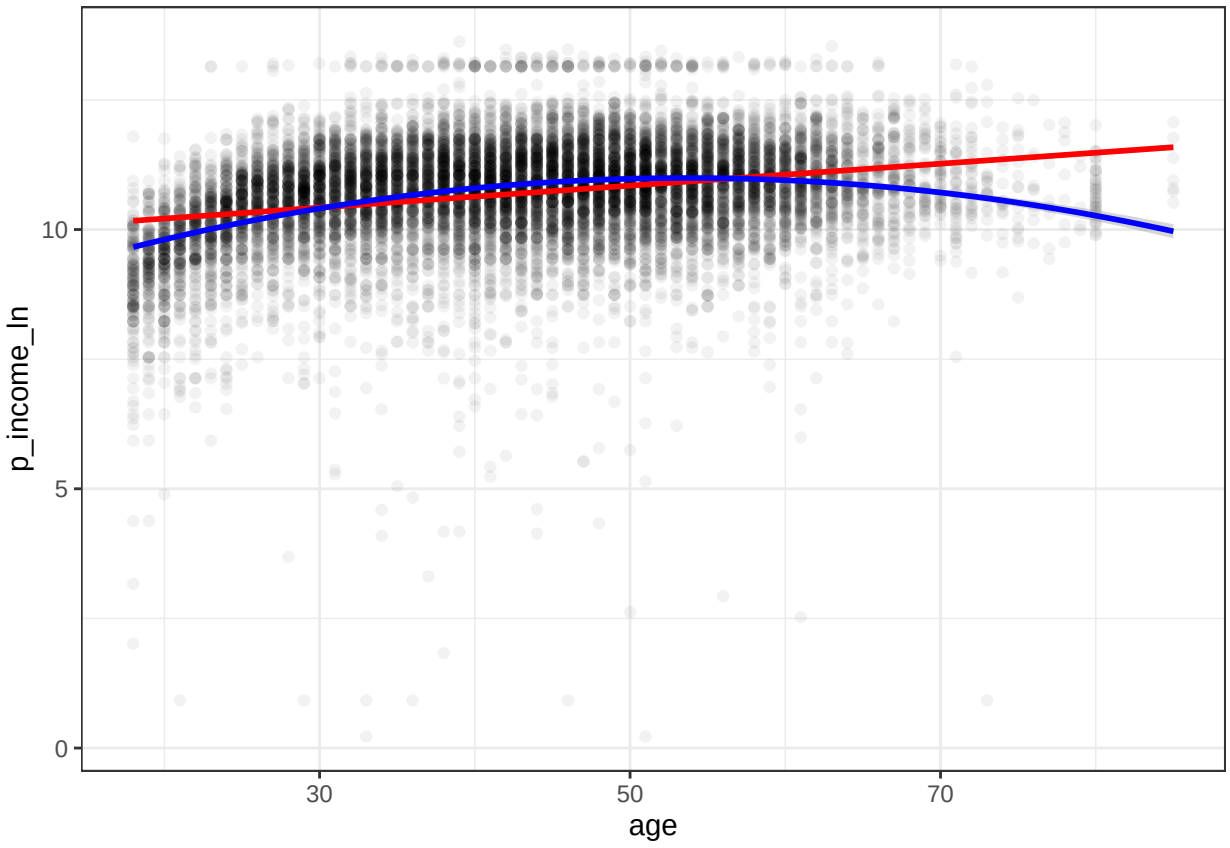



```
# Much easier to see the relationship and a clear pattern now. Does including
# a squared term seem to help?

mod2 <- lm(p_income_ln ~ age + I(age^2), df_plot_subset)
summary(mod2)
```

```
##
## Call:
## lm(formula = p_income_ln ~ age + I(age^2), data = df_plot_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7631  -0.4541   0.0798   0.5668   3.1354
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.980e+00  7.644e-02  104.39   <2e-16 ***
## age          1.123e-01  3.632e-03   30.92   <2e-16 ***
```

```
## I(age^2)    -1.047e-03  4.106e-05  -25.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9583 on 13510 degrees of freedom
## Multiple R-squared:  0.1151, Adjusted R-squared:  0.115
## F-statistic: 878.9 on 2 and 13510 DF,  p-value: < 2.2e-16
```

```
df_plot_subset[, preds1 := predict(mod2)]

ggplot(df_plot_subset, aes(y = p_income_ln, x = age)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method='lm', formula = y ~ x, color = "red") +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), color = "blue") +
  theme_bw()
```



It looks like we still see a similar pattern after we condition on employment status and use logged income. The second model seems to fit the pattern in the data a little bit better initially but seems to do worse for income in the older age groups. The question here would be: should we be analyzing this pattern for people working in late age? Or really, how can we improve the fit of this model further to capture the observed patterns in the dataset?

Data analysis

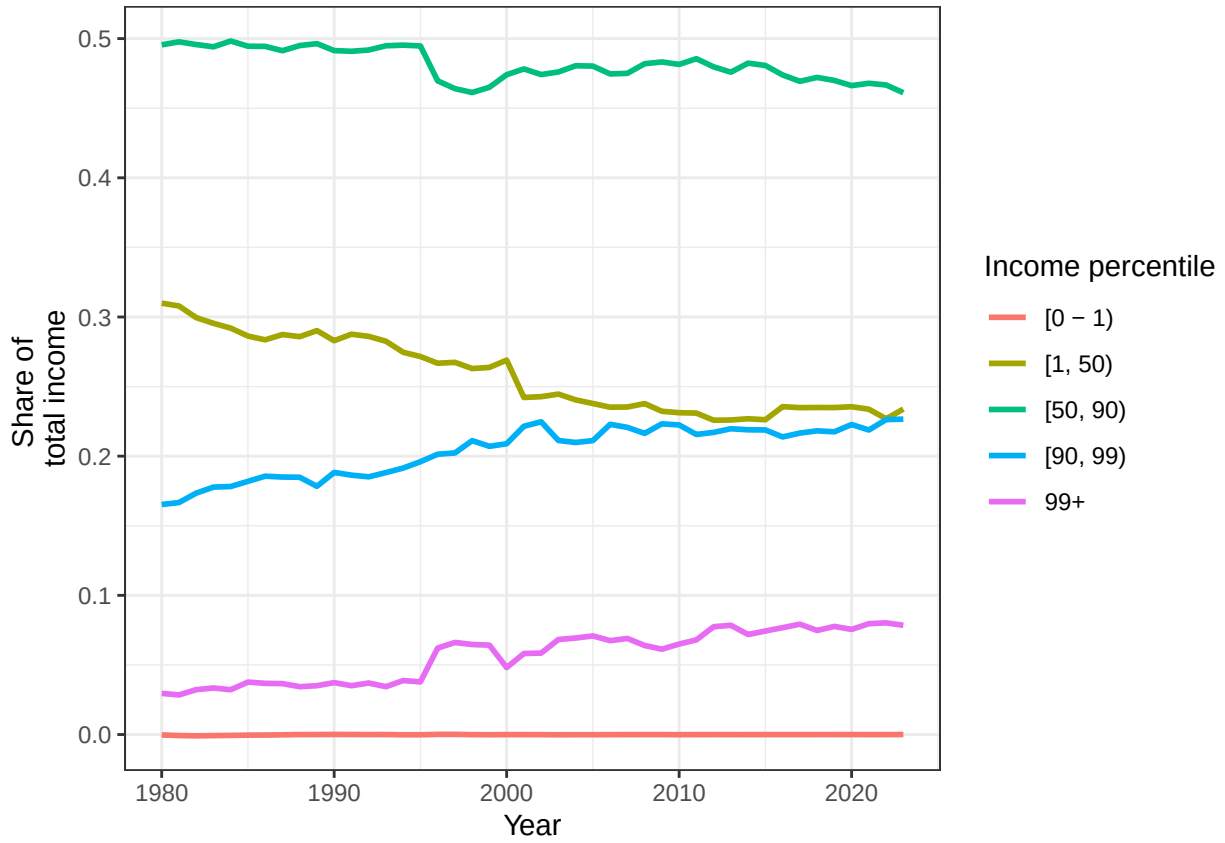
Let's start by trying to replicate the findings in FRED.

```r
# Let's calculate the share of total income by each percentile group:
cps_share <- cps[pid == 1, .(year, hh_income_group, hh_income)]
cps_share[, income_total := sum(.SD$hh_income), by = "year"]
cps_share[, income_percentile := sum(.SD$hh_income),
          by = c("year", "hh_income_group")]

cps_share[, income_share := income_percentile/income_total]

cps_share <- unique(cps_share[, .(year, hh_income_group, income_share)])

# Sanity check: Does the share of income sum to 1 each year?
cps_share[, sum(.SD$income_share), by = "year"]
```

```
##     year V1
##  1: 1980  1
##  2: 1981  1
##  3: 1982  1
##  4: 1983  1
##  5: 1984  1
##  6: 1985  1
##  7: 1986  1
##  8: 1987  1
##  9: 1988  1
## 10: 1989  1
## 11: 1990  1
## 12: 1991  1
## 13: 1992  1
## 14: 1993  1
## 15: 1994  1
## 16: 1995  1
## 17: 1996  1
## 18: 1997  1
## 19: 1998  1
## 20: 1999  1
## 21: 2000  1
## 22: 2001  1
## 23: 2002  1
## 24: 2003  1
## 25: 2004  1
## 26: 2005  1
## 27: 2006  1
## 28: 2007  1
## 29: 2008  1
## 30: 2009  1
## 31: 2010  1
## 32: 2011  1
## 33: 2012  1
## 34: 2013  1
## 35: 2014  1
## 36: 2015  1
## 37: 2016  1
## 38: 2017  1
## 39: 2018  1
```

```
## 40: 2019  1
## 41: 2020  1
## 42: 2021  1
## 43: 2022  1
## 44: 2023  1
##     year V1
```

```
ggplot(cps_share, aes(x = year, y = income_share, color = hh_income_group)) +
  geom_line(size = 1) +
  labs(x = "Year", y = "Share of \ntotal income",
       color = "Income percentile") +
  theme_bw()
```

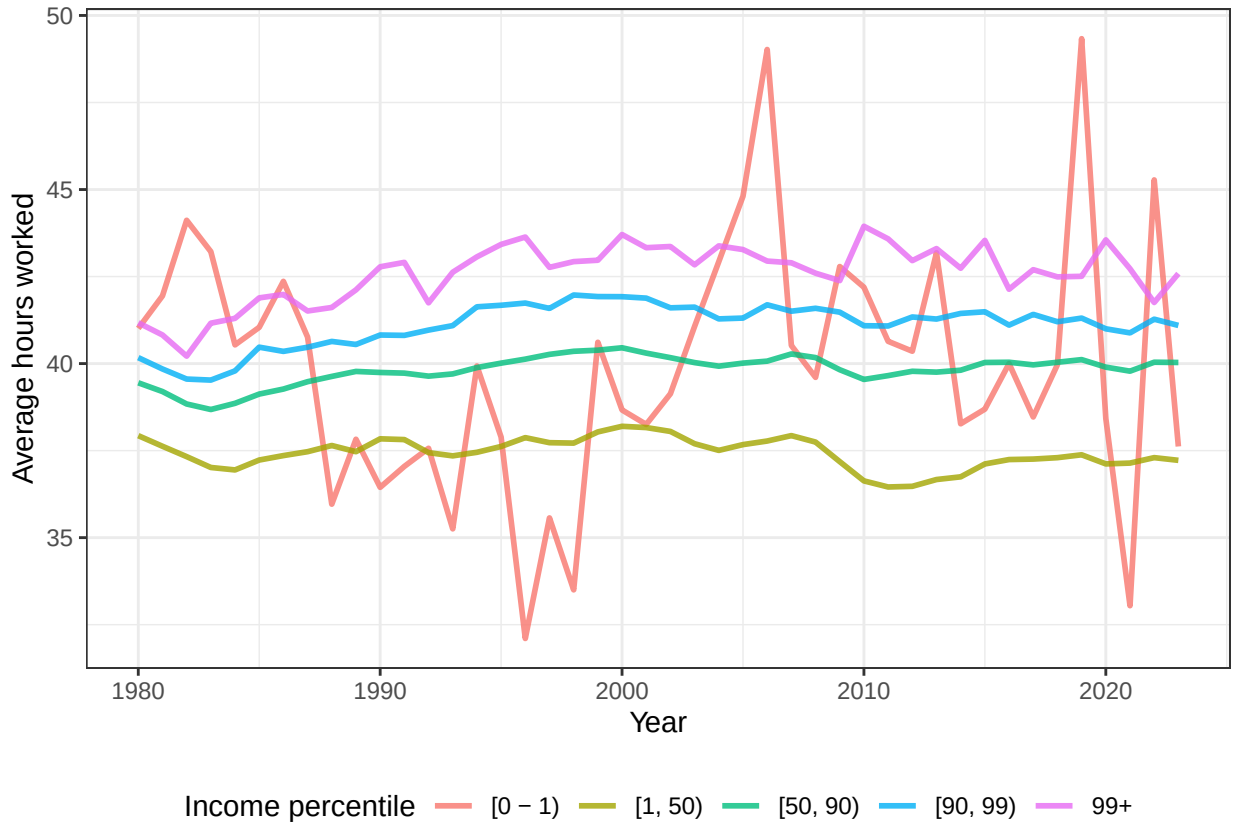


We're not able to even come close to the dramatic findings in FRED, which indicates any analysis we do will likely be limited by the CPS sample. The biggest issue here is that people in the 99th percentile are the least likeliest to be sampled within CPS. So simply adding up income in CPS doesn't even come close to capturing the trends found using IRS data. While some of the general features are still there - income share in the 1st - 50th percentile is decreasing - we see an inversion in the 90 to 99th percentile compared to the FRED findings, and the decrease in the 50 - 90 percentile is really marginal compared to what's happening in reality.

Let's try seeing if there's been a change in the hours worked.

```
# Let's calculate the share of total income by each percentile group:
cps[, mean_hours_percentile := mean(.SD$hours_per_week, na.rm = T),
    by = c("year", "hh_income_group")]
```

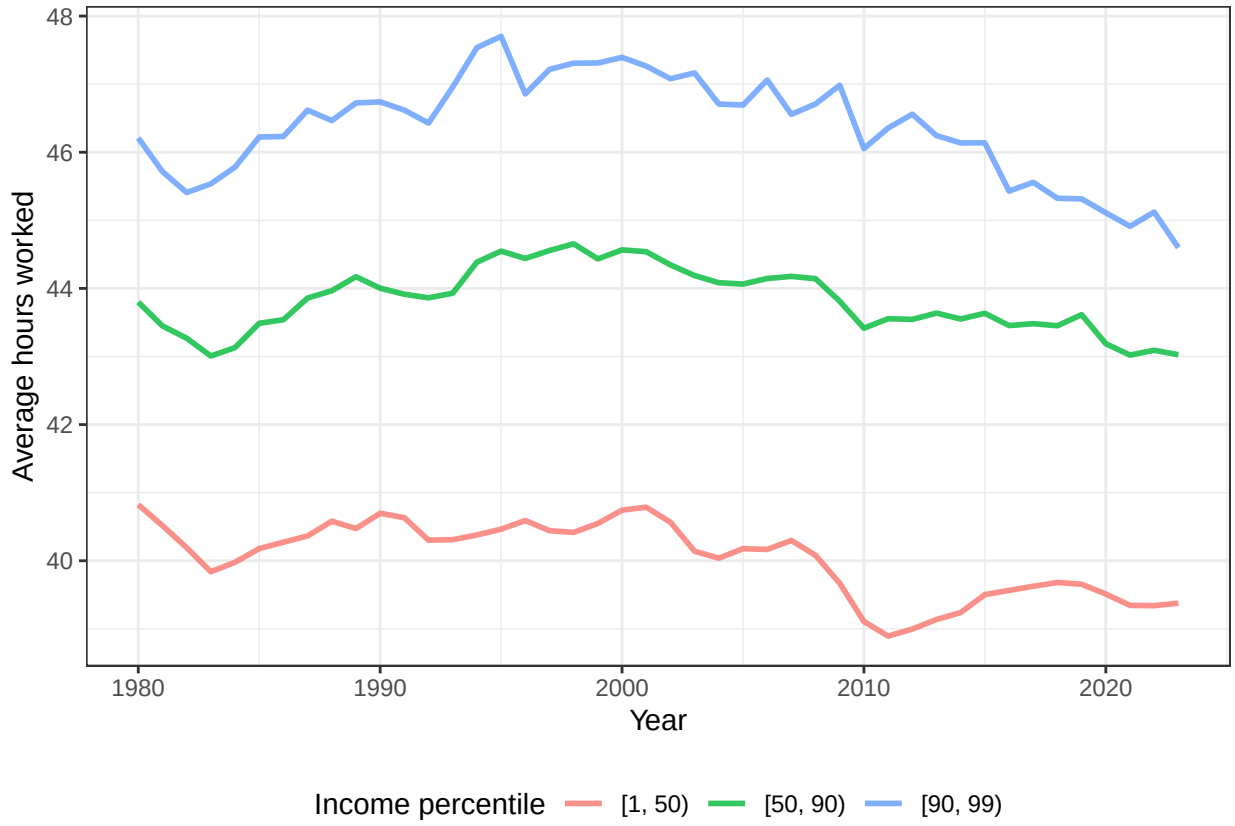
```
df_plot <- unique(cps[, .(year, hh_income_group, mean_hours_percentile)])

ggplot(df_plot, aes(x = year, y = mean_hours_percentile, color = hh_income_group)) +
  geom_line(size = 1, alpha = 0.8) +
  labs(x = "Year", y = "Average hours worked",
       color = "Income percentile") +
  theme_bw() +
  theme(legend.position = "bottom")
```



This initial graph doesn't look too promising and there's likely tons of reasons why this graph is ultimately inadequate at addressing the original research question. For one, this is plotting mean hours worked by household income percentile. However, for a two or more earner household, there might be a primary earner, along with several individuals who work less. So we might only want to look at the primary household earner to calculate this statistic. It also looks like there's probably an insufficient amount of data to look at the lowest income bracket (or the highest). We also need to restrict the sample to people who are employed.

```
# Determine who earns the most personal income in a household, then
# create a dummy variable for that individual. Subset to these observations
cps[, top_earner := max(p_income), by = "hhid"]
cps[, top_earner := ifelse(p_income == top_earner, 1, 0), by = "hhid"]

# Restrict sample to the top earners
df_plot <- cps[top_earner == 1 & employment_status == "employed", ]

# Remove the 1st and 99th percentile:
```

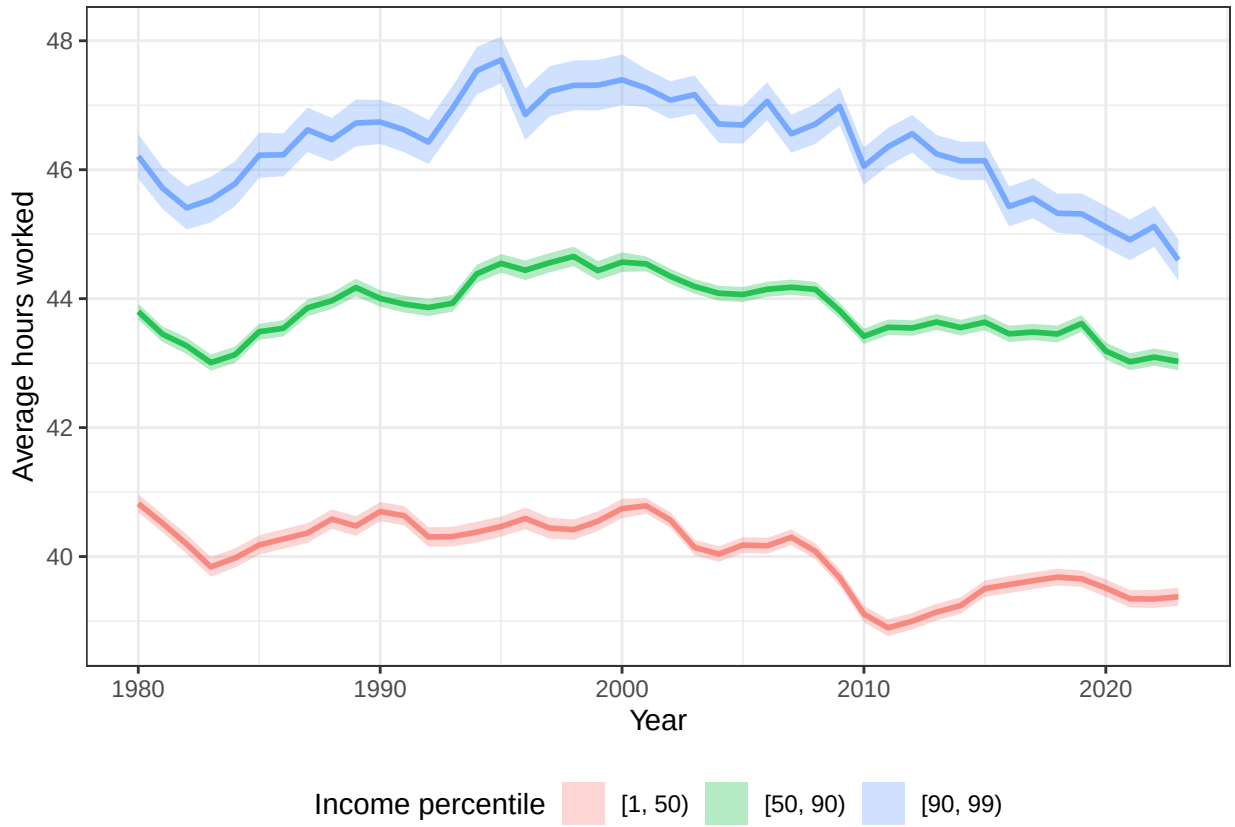
```
df_plot <- df_plot[hh_income_group != "[0 - 1)" & hh_income_group != "99+"]

df_plot[, mean_hours_percentile := mean(.SD$hours_per_week, na.rm = T),
        by = c("year", "hh_income_group")]

df_plot <- unique(df_plot[, .(year, hh_income_group, mean_hours_percentile)])

ggplot(df_plot, aes(x = year, y = mean_hours_percentile, color = hh_income_group)) +
  geom_line(size = 1, alpha = 0.8) +
  labs(x = "Year", y = "Average hours worked",
       color = "Income percentile") +
  theme_bw() +
  theme(legend.position = "bottom")
```



There's still a ton of variablity from year to year so let's add in a rough measure of uncertainty to better understand if there's a meaningful difference across these groups or by year.

```
df_plot <- cps[top_earner == 1 & employment_status == "employed", ]
df_plot <- df_plot[hh_income_group != "[0 - 1)" & hh_income_group != "99+"]

df_plot[, mean_hours_percentile := mean(.SD$hours_per_week, na.rm = T),
        by = c("year", "hh_income_group")]
df_plot[, mean_hours_sd := sd(.SD$hours_per_week, na.rm = T),
        by = c("year", "hh_income_group")]

df_plot[, n_sample := .N, by = c("year", "hh_income_group")]
```

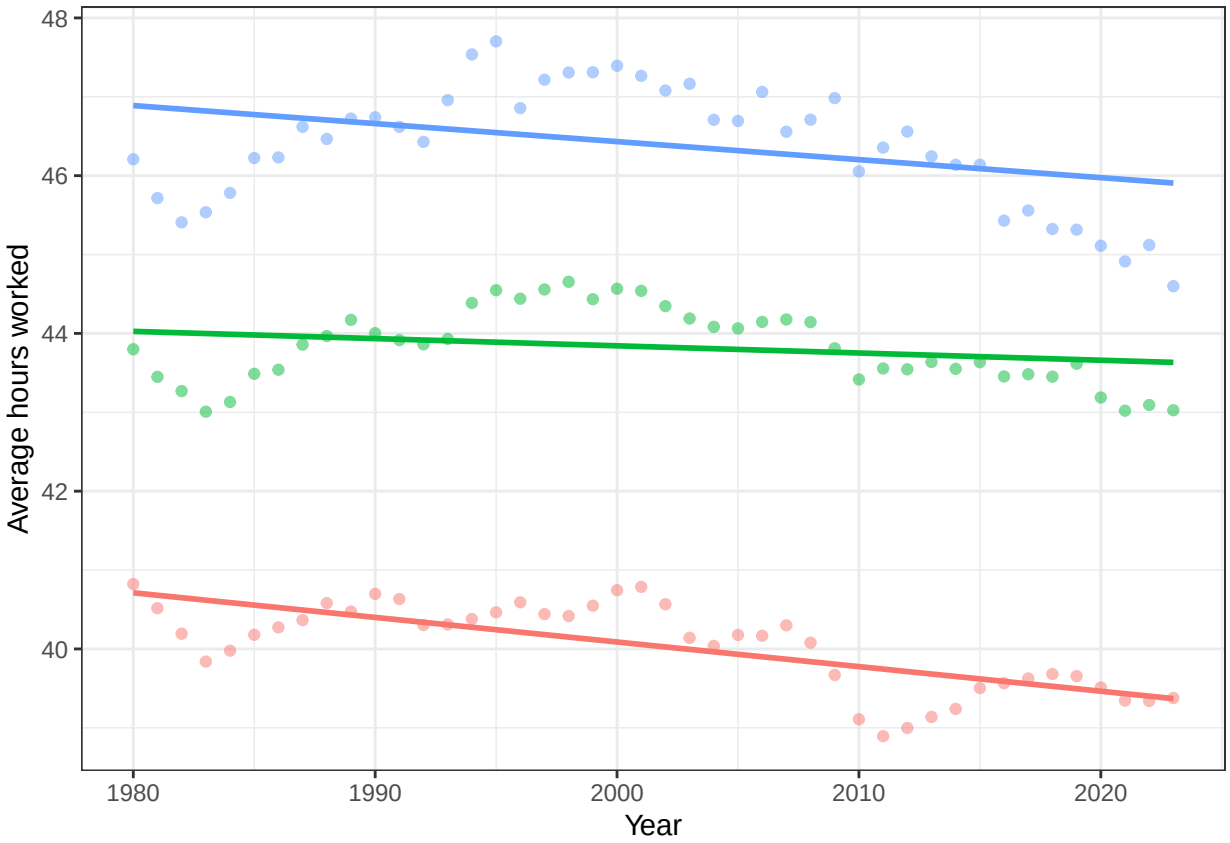
```
df_plot[, mean_hours_lower := mean_hours_percentile - 1.96*mean_hours_sd/sqrt(n_sample)]
df_plot[, mean_hours_upper := mean_hours_percentile + 1.96*mean_hours_sd/sqrt(n_sample)]

df_plot <- unique(df_plot[, .(year, hh_income_group, mean_hours_percentile,
                              mean_hours_lower, mean_hours_upper)])

ggplot(df_plot, aes(x = year, y = mean_hours_percentile)) +
        geom_line(size = 1, alpha = 0.8, aes(color = hh_income_group)) +
        geom_ribbon(aes(ymin = mean_hours_lower, ymax = mean_hours_upper,
                        fill = hh_income_group), alpha = 0.3) +
        labs(x = "Year", y = "Average hours worked",
             fill = "Income percentile") +
        guides(color = "none") +
        theme_bw() +
        theme(legend.position = "bottom")
```



Intervals are pretty tight on the means so based on CPS, it looks like if anything, hours have slightly decreased across income percentile groups. Let's see if we can confirm this with a model:

```
df_plot <- cps[top_earner == 1 & employment_status == "employed" &
                !is.na(hours_per_week), ]

df_plot <- df_plot[hh_income_group != "[0 - 1)" & hh_income_group != "99+"]

mod_hours <- lm(hours_per_week ~ year*hh_income_group, data = df_plot)
```

```r
pred_data <- setDT(expand.grid("year" = 1980:2023,
                               "hh_income_group" = unique(df_plot$hh_income_group)))

pred_data[, preds := predict(mod_hours, newdata = pred_data)]

df_plot[, mean_hours_percentile := mean(.SD$hours_per_week, na.rm = T),
        by = c("year", "hh_income_group")]
df_plot[, mean_hours_sd := sd(.SD$hours_per_week, na.rm = T),
        by = c("year", "hh_income_group")]

df_plot[, n_sample := .N, by = c("year", "hh_income_group")]

df_plot[, mean_hours_lower := mean_hours_percentile - 1.96*mean_hours_sd/sqrt(n_sample)]
df_plot[, mean_hours_upper := mean_hours_percentile + 1.96*mean_hours_sd/sqrt(n_sample)]

df_plot <- unique(df_plot[, .(year, hh_income_group, mean_hours_percentile,
                              mean_hours_lower, mean_hours_upper)])

ggplot(df_plot, aes(x = year, y = mean_hours_percentile, color = hh_income_group)) +
        geom_point(alpha = 0.5) +
        geom_line(data = pred_data, aes(y = preds), size = 1) +
        labs(x = "Year", y = "Average hours worked",
             fill = "Income percentile") +
        guides(color = "none") +
        theme_bw() +
        theme(legend.position = "bottom")
```



Probably need to build a little more flexibility into the above model to capture some of the observed trends, though these trends might be ephemeral. I would need to investigate this further, considering additional relationships across some of the other variables. Small multiple plots would help with this.