#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# PROJECT: KILLEWALD- GENDER WAGE GAP

# PSID RAW TO SAMPLE

# AUTHOR: NINO CRICCO

# LAST UPDATED: 10/13/2020 (mdy)

# RUNTIME: 1131.871 sec (~19 min)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# https://www.bls.gov/cps/cenocc2010.htm

# LOADING LIBRARIES

library(tidyverse)

library(modelr)

library(spatstat)

library(xtable)

library(stargazer)

library(weights)

library(knitr)

library(gridExtra)

library(McSpatial)

library(RColorBrewer)

library(skimr)

library(readr)pol

library(normalr)

library(fastDummies)

library(readstata13)

library(lubridate)

library(readxl)

library(janitor)

library(grid)

library(crosswalkr)

library(foreign)

library(scales)

library(reshape2)

library(haven)

library(foreach)

library(lmtest)

library(broom)

library(tictoc)

tic()

# Loading helper functions

source("Jobs/helperfunctions.R")

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# LOADING AND CLEANING DATA

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# First loading both crosswalks for occupation and industry codes,

# gathered from IPUMS at the following link at time of writing:

R# https://usa.ipums.org/usa/resources/volii/documents/integrated\_ind\_occ\_crosswalks.xlsx

crosswalk.occ <- read\_xlsx("Raw Data/crosswalks/ipums\_occ\_crosswalk.xlsx") %>%

rename(occ2010 = "OCC2010",

occlab = 'Occupation category description',

occ1970 = '1970',

occ2000 = '2000',

acs2010 = "ACS2010")

crosswalk.ind <- read\_xlsx("Raw Data/crosswalks/ipums\_industry\_crosswalk.xlsx") %>%

rename(ind1990 = IND1990,

occlab = 'Industry category description',

ind1970 = '1970',

ind2000 = '2000')

crosswalk.ind.2012 <- read\_xlsx("Raw Data/crosswalks/census\_industry\_crosswalk\_2012.1990.xlsx") %>%

rename(ind1990 = "1990 Census",

occlab = "Census 2012 Category Title",

ind2012 = "2012 Census Code")

# Generating data

# Both gathers values from the PSID raw file and transforms/recodes values in the process

psid\_raw <- read.dta("Raw Data/psid/psid\_wrk.dta") %>%

transmute(

family\_id = ER30001, # 1968 interview number

person\_number = ER30002, # 1968 person number

samp\_error\_stratum = ER31996, # PSID Sampling error variable- strata

samp\_error\_cluster = ER31997, # PSID Sampling error variable- cluster

indiv.id = paste(family\_id, person\_number, sep ="\_"), # unique individual identifier

female = ifelse(ER32000 == 1, 0, ifelse(ER32000 == 2, 1, NA)), # identifying sex

year.firstbornchild = ifelse(ER32024 == 9999, NA, ER32024),

yr.born\_2017 = ifelse(ER34506 %in% c(0, 9999), NA, ER34506),

int.num\_1980 = ER30313, int.num\_1981 = ER30343, int.num\_1990 = ER30642, int.num\_1991 = ER30689,

int.num\_1999 = ER33501, int.num\_2001 = ER33601, int.num\_2009 = ER34001, int.num\_2011 = ER34101,

int.num\_2015 = ER34301, int.num\_2017 = ER34501, int.num\_2019 = ER34701, # Year int number

seq.num\_1980 = ER30314, seq.num\_1981 = ER30344, seq.num\_1990 = ER30643, seq.num\_1991 = ER30690,

seq.num\_1999 = ER33502, seq.num\_2001 = ER33602, seq.num\_2009 = ER34002, seq.num\_2011 = ER34102,

seq.num\_2015 = ER34302, seq.num\_2017 = ER34502, seq.num\_2019 = ER34702, # Year seq number

perwt\_1980 = ER30342, perwt\_1981 = ER30372, perwt\_1990 = ER30686, perwt\_1991 = ER30730, # for 1990-91, using weights that exclude the Latino sample

perwt\_1999 = ER33546, perwt\_2001 = ER33637, perwt\_2009 = ER34045, perwt\_2011 = ER34154,

perwt\_2015 = ER34413, perwt\_2017 = ER34651, perwt\_2017 = ER34651, perwt\_2019 = ER34864,# individual level weights

# For perwt.imm vars: same weights for all years but 2017, in 2017 excluding 2017 immigrant sample

perwt.imm\_1980 = ER30342, perwt.imm\_1981 = ER30372, perwt.imm\_1990 = ER30686, perwt.imm\_1991 = ER30730,

perwt.imm\_1999 = ER33546, perwt.imm\_2001 = ER33637, perwt.imm\_2009 = ER34045, perwt.imm\_2011 = ER34154,

perwt.imm\_2015 = ER34413, perwt.imm\_2017 = ER34650, perwt.imm\_2019 = ER34863, # individual level weights, excluding immigrant sample in 2017

# For perwt.lat vars: same weights for all years but 1990-91, in 90s including latino sample

perwt.lat\_1980 = ER30342, perwt.lat\_1981 = ER30372, perwt.lat\_1990 = ER30688, perwt.lat\_1991 = ER30732,

perwt.lat\_1999 = ER33546, perwt.lat\_2001 = ER33637, perwt.lat\_2009 = ER34045, perwt.lat\_2011 = ER34154,

perwt.lat\_2015 = ER34413, perwt.lat\_2017 = ER34651, perwt.lat\_2019 = ER34864,

# For perwt.lat,imm vars: same weights for all years but 1990-91, in 90s including latino sample, in 2017 excludes imm sample

perwt.lat.imm\_1980 = ER30342, perwt.lat.imm\_1981 = ER30372, perwt.lat.imm\_1990 = ER30688, perwt.lat.imm\_1991 = ER30732,

perwt.lat.imm\_1999 = ER33546, perwt.lat.imm\_2001 = ER33637, perwt.lat.imm\_2009 = ER34045, perwt.lat.imm\_2011 = ER34154,

perwt.lat.imm\_2015 = ER34413, perwt.lat.imm\_2017 = ER34650, perwt.lat.imm\_2019 = ER34863, # individual level weights, excluding immigrant sample in 2017

# Relationship to head- coding as head/RP, wife/SP, other

rel.head\_1980 = case\_when(ER30315 == 1 ~ "head", ER30315 == 2 ~ "wife", TRUE ~ "other"),

rel.head\_1981 = case\_when(ER30345 == 1 ~ "head", ER30345 == 2 ~ "wife", TRUE ~ "other"),

rel.head\_1990 = case\_when(ER30644 == 10 ~ "head", ER30691 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_1991 = case\_when(ER30691 == 10 ~ "head", ER30691 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_1999 = case\_when(ER33503 == 10 ~ "head", ER33603 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2001 = case\_when(ER33603 == 10 ~ "head", ER33603 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2009 = case\_when(ER34003 == 10 ~ "head", ER34103 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2011 = case\_when(ER34103 == 10 ~ "head", ER34103 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2015 = case\_when(ER34303 == 10 ~ "head", ER34303 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2017 = case\_when(ER34503 == 10 ~ "head", ER34503 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

rel.head\_2019 = case\_when(ER34703 == 10 ~ "head", ER34703 %in% c(20, 22) ~ "wife", TRUE ~ "other"),

# Age- setting codes 0, 999 to NA

age\_1980 = na\_codes(ER30316, c(0, 999)), age\_1981 = na\_codes(ER30346, c(0, 999)),

age\_1990 = na\_codes(ER30645, c(0, 999)), age\_1991 = na\_codes(ER30692, c(0, 999)),

age\_1999 = na\_codes(ER33504, c(0, 999)), age\_2001 = na\_codes(ER33604, c(0, 999)),

age\_2009 = na\_codes(ER34004, c(0, 999)), age\_2011 = na\_codes(ER34104, c(0, 999)),

age\_2015 = na\_codes(ER34305, c(0, 999)), age\_2017 = na\_codes(ER34504, c(0,999)),

age\_2019 = na\_codes(ER34704, c(0,999)),

# Race

# For the early years, we recode a single measure of race, which

# is carried over from the 1972 and there is a single value assinged to the family. For sample members

# who split after 1972, members are assigned the race in the PSID family in 1972

racehd\_1980 = case\_when(V7447 == 1 ~ "White", V7447 == 2 ~ "Black", V7447 == 3 ~ "Hispanic", V7447 == 7 ~ "Other"),

racehd\_1981 = case\_when(V8099 == 1 ~ "White", V8099 == 2 ~ "Black", V8099 == 3 ~ "Hispanic", V8099 == 7 ~ "Other"),

racehd\_1990 = NA, racehd\_1991 = NA, racehd\_1999 = NA, racehd\_2001 = NA, racehd\_2009 = NA, racehd\_2011 = NA,

# Marital status of head

marstat.hd\_1980 = V7261, marstat.hd\_1981 = V7952, marstat.hd\_1990 = V18055, marstat.hd\_1991 = V19355,

marstat.hd\_1999 = ER13021, marstat.hd\_2001 = ER17024,

marstat.hd\_2009 = na\_codes(ER42023, c(8,9)), marstat.hd\_2011 = na\_codes(ER47323, c(8,9)),

marstat.hd\_2015 = na\_codes(ER60024, c (8,9)), marstat.hd\_2017 = na\_codes(ER66024, c(8,9)),

marstat.hd\_2019 = na\_codes(ER72024, c(8,9)),

# Number of kids in the family unit

numkids.fu\_1980 = V7070, numkids.fu\_1981 = V7661, numkids.fu\_1990 = V18052, numkids.fu\_1991 = V19352,

numkids.fu\_1999 = ER13013, numkids.fu\_2001 = ER17016, numkids\_2009 = ER42020, numkids.fu\_2011 = ER47320,

numkids.fu\_2015 = ER60021, numkids.fu\_2017 = ER66021, numkids.fu\_2019 = ER72021,

# Region

region\_1980 = na\_if(V7419, 9), region\_1981 = na\_if(V8071, 9),

region\_1990 = na\_if(V18889, 9), region\_1991 = na\_if(V20189, 9),

region\_1999 = na\_if(ER16430, 9), region\_2001 = na\_if(ER20376, 9),

region\_2009 = na\_if(ER46974, 9), region\_2011 = na\_if(ER52398, 9),

region\_2015 = na\_if(ER65451, 9), region\_2017 = na\_if(ER71530, 9),

region\_2019 = na\_if(ER77591, 9),

# Education (taken from the family files- missing less data than in the individual files)

yrs.ed.fam\_hd\_1980 = na\_if(V7387, 99), # Here zero is no educ

yrs.ed.fam\_wf\_1980 = na\_codes(V7346, c(0, 99)), # Here zero is no educ or no wife

yrs.ed.fam\_hd\_1981 = na\_if(V8039, 99), yrs.ed.fam\_wf\_1981 = na\_codes(V7998, c(0, 99)),

yrs.ed.fam\_hd\_1990 = na\_if(V18898, 9), yrs.ed.fam\_wf\_1990 = na\_codes(V18899, c(0, 9)), # 1990: educ codes are different

yrs.ed.fam\_hd\_1991 = na\_if(V20198, 99), yrs.ed.fam\_wf\_1991 = na\_codes(V20199, c(0, 99)),

yrs.ed.fam\_hd\_1999 = na\_if(ER16516, 99), yrs.ed.fam\_wf\_1999 = na\_codes(ER16517, c(0, 99)),

yrs.ed.fam\_hd\_2001 = na\_if(ER20457, 99), yrs.ed.fam\_wf\_2001 = na\_codes(ER20458, c(0, 99)),

yrs.ed.fam\_hd\_2009 = na\_if(ER46981, 99), yrs.ed.fam\_wf\_2009 = na\_codes(ER46982, c(0, 99)),

yrs.ed.fam\_hd\_2011 = na\_if(ER52405, 99), yrs.ed.fam\_wf\_2011 = na\_codes(ER52406, c(0, 99)),

yrs.ed.fam\_hd\_2015 = na\_if(ER65459, 99), yrs.ed.fam\_wf\_2015 = na\_codes(ER65460, c(0, 99)),

yrs.ed.fam\_hd\_2017 = na\_if(ER71538, 99), yrs.ed.fam\_wf\_2017 = na\_codes(ER71539, c(0, 99)),

yrs.ed.fam\_hd\_2019 = na\_if(ER77599, 99), yrs.ed.fam\_wf\_2019 = na\_codes(ER77600, c(0, 99)),

# Advanced degrees

# For early years, coding has degree as 1, No (5) or Inappropriate (0) as 0, implicit 9 values are converted to NA

ba\_hd\_1980 = case\_when(V7393 == 1 ~ 1, V7393 %in% c(0,5) ~ 0),

ba\_wf\_1980 = case\_when(V7349 == 1 ~ 1, V7349 %in% c(0,5) ~ 0),

ba\_hd\_1981 = case\_when(V8045 == 1 ~ 1, V8045 %in% c(0,5) ~ 0),

ba\_wf\_1981 = case\_when(V8001 == 1 ~ 1, V8001 %in% c(0,5) ~ 0),

advdeg\_hd\_1980 = case\_when(V7394 == 1 ~ 1, V7394 %in% c(0,5) ~ 0),

advdeg\_wf\_1980 = case\_when(V7350 == 1 ~ 1, V7350 %in% c(0,5) ~ 0),

advdeg\_hd\_1981 = case\_when(V8046 == 1 ~ 1, V8046 %in% c(0,5) ~ 0),

advdeg\_wf\_1981 = case\_when(V8002 == 1 ~ 1, V8002 %in% c(0,5) ~ 0),

# For later years, using highest degree earned vars following Blau & Kahn for categorization of advanced degrees

ba\_hd\_1990 = ifelse(V18833 == 2, 1, 0), ba\_wf\_1990 = ifelse(V20063 == 2, 1, 0),

advdeg\_hd\_1990 = ifelse(V18833 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_1990 = ifelse(V20063 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_1991 = ifelse(V20133 == 2, 1, 0), ba\_wf\_1991 = ifelse(V20063 == 2, 1, 0),

advdeg\_hd\_1991 = ifelse(V20133 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_1991 = ifelse(V20063 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_1999 = ifelse(ER15953 == 2, 1, 0), ba\_wf\_1999 = ifelse(ER15861 == 2, 1, 0),

advdeg\_hd\_1999 = ifelse(ER15953 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_1999 = ifelse(ER15861 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2001 = ifelse(ER20014 == 2, 1, 0), ba\_wf\_2001 = ifelse(ER19922 == 2, 1, 0),

advdeg\_hd\_2001 = ifelse(ER20014 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2001 = ifelse(ER19922 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2009 = ifelse(ER46568 == 2, 1, 0), ba\_wf\_2009 = ifelse(ER46474 == 2, 1, 0),

advdeg\_hd\_2009 = ifelse(ER46568 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2009 = ifelse(ER46474 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2011 = ifelse(ER51929 == 2, 1, 0), ba\_wf\_2011 = ifelse(ER51835 == 2, 1, 0),

advdeg\_hd\_2011 = ifelse(ER51929 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2011 = ifelse(ER51835 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2015 = ifelse(ER64837 == 2, 1, 0), ba\_wf\_2015 = ifelse(ER64698 == 2, 1, 0),

advdeg\_hd\_2015 = ifelse(ER64698 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2015 = ifelse(ER64837 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2017 = ifelse(ER70909 == 2, 1, 0), ba\_wf\_2017 = ifelse(ER51835 == 2, 1, 0),

advdeg\_hd\_2017 = ifelse(ER70909 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2017 = ifelse(ER51835 %in% c(3, 4, 5, 6), 1, 0),

ba\_hd\_2019 = ifelse(ER76924 == 2, 1, 0), ba\_wf\_2019 = ifelse(ER76779 == 2, 1, 0),

advdeg\_hd\_2019 = ifelse(ER76924 %in% c(3, 4, 5, 6), 1, 0), advdeg\_wf\_2019 = ifelse(ER76779 %in% c(3, 4, 5, 6), 1, 0),

# Total annual work hours- recorded in a given year about prior year,

# so 1981 values reflect hours worked in 1980, 2017 values reflect

# hours worked in 2016. 0 = did not work for money. Missing values assigned by PSID

ann.wrk.hrs\_hd\_1980 = V6934, ann.wrk.hrs\_hd\_1981 = V7530,

ann.wrk.hrs\_wf\_1980 = V6946, ann.wrk.hrs\_wf\_1981 = V7540,

ann.wrk.hrs\_hd\_1990 = V17744, ann.wrk.hrs\_wf\_1990 = V17774,

ann.wrk.hrs\_hd\_1991 = V19044, ann.wrk.hrs\_wf\_1991 = V19074,

ann.wrk.hrs\_hd\_1999 = ER16471, ann.wrk.hrs\_wf\_1999 = ER16482,

ann.wrk.hrs\_hd\_2001 = ER20399, ann.wrk.hrs\_wf\_2001 = ER20410,

ann.wrk.hrs\_hd\_2009 = ER46767, ann.wrk.hrs\_wf\_2009 = ER46788,

ann.wrk.hrs\_hd\_2011 = ER52175, ann.wrk.hrs\_wf\_2011 = ER52196,

ann.wrk.hrs\_hd\_2015 = ER65156, ann.wrk.hrs\_hd\_2017 = ER71233, ann.wrk.hrs\_hd\_2019 = ER77255,

ann.wrk.hrs\_wf\_2015 = ER65177, ann.wrk.hrs\_wf\_2017 = ER71254, ann.wrk.hrs\_wf\_2019 = ER77276,

# Total weeks worked: recorded in year about prior year

# so 1981 values reflect weeks worked in 1980, 2017 values reflect

# weeks worked in 2016. 0 reflects inapp in early years (data not collected bc of respondent's employment status)

# and no weeks worked during the later years. For the later year values 0 reflects no weeks worked for head, no

# weeks worked or no spouse for spouse. Missinig data assigned by PSID

wks.wrk\_hd\_1980 = na\_if(V7118, 99), wks.wrk\_wf\_1980 = na\_if(V7213, 99),

wks.wrk\_hd\_1981 = na\_if(V7741, 99), wks.wrk\_wf\_1981 = na\_if(V7904, 99),

wks.wrk\_hd\_1990 = na\_if(V18196, 99), wks.wrk\_wf\_1990 = na\_if(V18498, 99),

wks.wrk\_hd\_1991 = na\_if(V19496, 99), wks.wrk\_wf\_1991 = na\_if(V19798, 99),

wks.wrk\_hd\_1999 = na\_codes(ER13362, c(97, 98, 99)), wks.wrk\_wf\_1999 = na\_codes(ER13874, c(97, 98, 99)),

wks.wrk\_hd\_2001 = na\_codes(ER17391, c(97, 98, 99)), wks.wrk\_wf\_2001 = na\_codes(ER17961, c(97, 98, 99)),

wks.wrk\_hd\_2009 = ER46761, wks.wrk\_wf\_2009 = ER46782,

wks.wrk\_hd\_2011 = ER52169, wks.wrk\_wf\_2011 = ER52190,

wks.wrk\_hd\_2015 = ER65150, wks.wrk\_wf\_2015 = ER65171,

wks.wrk\_hd\_2017 = ER71227, wks.wrk\_wf\_2017 = ER71248,

wks.wrk\_hd\_2019 = ER77249, wks.wrk\_wf\_2019 = ER77270,

# Housework

# In early years, housework is top coded at 98: in later years, we manually topcode values >98 to 98 for consistency

housework\_hd\_1980 = na\_if(V7266, 99), housework\_wf\_1980 = na\_if(V7265, 99),

housework\_hd\_1981 = na\_if(V7957, 99), housework\_wf\_1981 = na\_if(V7956, 99),

housework\_hd\_1990 = na\_if(V18698, 99), housework\_wf\_1990 = na\_if(V18697, 99), # Note: topcode in 90-91 is 84

housework\_hd\_1991 = na\_if(V19998, 99), housework\_wf\_1991 = na\_if(V19997, 99),

housework\_hd\_1999 = na\_codes(ER14230, c(0.5, 998, 999)), housework\_hd\_1999 = case\_when(housework\_hd\_1999 >= 98 ~ 98, TRUE ~ housework\_hd\_1999),

housework\_wf\_1999 = na\_codes(ER14229, c(998, 999)), housework\_wf\_1999 = case\_when(housework\_wf\_1999 >= 98 ~ 98, TRUE ~ housework\_wf\_1999),

housework\_hd\_2001 = na\_codes(ER18359, c(998, 999)), housework\_hd\_2001 = case\_when(housework\_hd\_2001 >= 98 ~ 98, TRUE ~ housework\_hd\_2001),

housework\_wf\_2001 = na\_codes(ER18357, c(998, 999)), housework\_wf\_2001 = case\_when(housework\_wf\_2001 >= 98 ~ 98, TRUE ~ housework\_wf\_2001),

housework\_hd\_2009 = na\_codes(ER42646, c(998, 999)), housework\_hd\_2009 = case\_when(housework\_hd\_2009 >= 98 ~ 98, TRUE ~ housework\_hd\_2009),

housework\_wf\_2009 = na\_codes(ER42644, c(998, 999)), housework\_wf\_2009 = case\_when(housework\_wf\_2009 >= 98 ~ 98, TRUE ~ housework\_wf\_2009),

housework\_hd\_2011 = na\_codes(ER47964, c(998, 999)), housework\_hd\_2011 = case\_when(housework\_hd\_2011 >= 98 ~ 98, TRUE ~ housework\_hd\_2011),

housework\_wf\_2011 = na\_codes(ER47962, c(998, 999)), housework\_wf\_2011 = case\_when(housework\_wf\_2011 >= 98 ~ 98, TRUE ~ housework\_wf\_2011),

housework\_hd\_2015 = na\_codes(ER60691, c(998, 999)), housework\_hd\_2015 = case\_when(housework\_hd\_2015 >= 98 ~ 98, TRUE ~ housework\_hd\_2015),

housework\_wf\_2015 = na\_codes(ER60689, c(998, 999)), housework\_wf\_2015 = case\_when(housework\_wf\_2015 >= 98 ~ 98, TRUE ~ housework\_wf\_2015),

housework\_hd\_2017 = na\_codes(ER66714, c(998, 999)), housework\_hd\_2017 = case\_when(housework\_hd\_2017 >= 98 ~ 98, TRUE ~ housework\_hd\_2017),

housework\_wf\_2017 = na\_codes(ER66727, c(998, 999)), housework\_wf\_2017 = case\_when(housework\_wf\_2017 >= 98 ~ 98, TRUE ~ housework\_wf\_2017),

housework\_hd\_2019 = na\_codes(ER72718, c(998, 999)), housework\_hd\_2019 = case\_when(housework\_hd\_2019 >= 98 ~ 98, TRUE ~ housework\_hd\_2019),

housework\_wf\_2019 = na\_codes(ER72731, c(998, 999)), housework\_wf\_2019 = case\_when(housework\_wf\_2019 >= 98 ~ 98, TRUE ~ housework\_wf\_2019),

# Occupation

# Note that in 1980, the PSID did not retrospectively recode occupation and industry variables from their

# PSID-specified 1970 codes to 1970 census codes. We use the non-recoded 1970 PSID occupation values in 1980

# to generate a harmonized scheme of occupation dummies (more details in the appendix). We also use the

# recoded occ values (1980 occupations recoded to the census occ) for those recoded, which we use to

# merge to IPUMs occupation characteristics. We use a similar procedure for industry in 1980

# First removing missing values

occ\_hd\_1980 = na\_codes(V7100, c(0, 99)), occ\_wf\_1980 = na\_codes(V7198, c(0,99)),

occ.recode\_hd\_1980 = na\_if(V7100\_A, 999), occ.recode\_wf\_1980 = na\_codes(V7198\_A, c(0, 999)),

# We use the recoded values and an IPUMS-provided crosswalk to convert the 1970 census occupation codes

# for heads and wives to 2010 occupation codes.

# We recode 600 (current armed forces), which has no match in 2010 occs, to 580 (military, rank not specified)

# We also reecode 329 (clerical and kindred occupation) to 325 (file clerks)

occ.recode\_hd\_1980 = case\_when(occ.recode\_hd\_1980 == 600 ~ 580, occ.recode\_hd\_1980 == 329 ~ 329, TRUE ~ occ.recode\_hd\_1980),

occ.recode\_wf\_1980 = case\_when(occ.recode\_wf\_1980 == 600 ~ 580, occ.recode\_wf\_1980 == 329 ~ 329, TRUE ~ occ.recode\_wf\_1980),

# This variable gets the crosswalk row number for rows that match a 1970 value to a 2010 value

occ.cw.position\_hd\_1980 = ifelse(is.na(occ.recode\_hd\_1980), NA, match(occ.recode\_hd\_1980, crosswalk.occ$occ1970, nomatch = NA)),

occ.cw.position\_wf\_1980 = ifelse(is.na(occ.recode\_wf\_1980), NA, match(occ.recode\_wf\_1980, crosswalk.occ$occ1970, nomatch = NA)),

# This variable codes a 2010 value to the 2010 value in the crosswalk that matches the crosswalk's row number

# Note: the crosswalk was edited to match "0" with "99998". Zero values indicate that the respondent was not

# eligible for the PSID's retroactive recoding

occ2010\_hd\_1980 = ifelse(is.na(occ.recode\_hd\_1980), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_1980]),

occ2010\_wf\_1980 = ifelse(is.na(occ.recode\_wf\_1980), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_1980]),

# For 1980- 1991, occupation values are coded into 1970 codes. We repeat the same procedure to create a variable that

# recodes 1970 census occupations observed in 1981 to 2010 census occupations in 1981, using the IPUMS crosswalk

occ\_hd\_1981 = ifelse(V7712 == 600, 580, V7712), occ\_hd\_1981 = na\_codes(occ\_hd\_1981, c(999, 0)),

occ.cw.position\_hd\_1981 = ifelse(is.na(occ\_hd\_1981), NA, match(occ\_hd\_1981, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_hd\_1981 = ifelse(is.na(occ\_hd\_1981 ), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_1981]),

occ\_wf\_1981 = ifelse(V7885 == 600, 580, V7885), occ\_wf\_1981 = na\_codes(occ\_wf\_1981, c(999, 0)),

occ.cw.position\_wf\_1981 = ifelse(is.na(occ\_wf\_1981), NA, match(occ\_wf\_1981, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_wf\_1981 = ifelse(is.na(occ\_wf\_1981), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_1981]),

occ\_hd\_1990 = ifelse(V18101 == 600, 580, V18101), occ\_hd\_1990 = na\_codes(occ\_hd\_1990, c(999, 0)),

occ.cw.position\_hd\_1990 = ifelse(is.na(occ\_hd\_1990), NA, match(occ\_hd\_1990, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_hd\_1990 = ifelse(is.na(occ\_hd\_1990 ), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_1990]),

occ\_wf\_1990 = ifelse(V18403 == 600, 580, V19703), occ\_wf\_1990 = na\_codes(occ\_wf\_1990, c(999, 0)),

occ.cw.position\_wf\_1990 = ifelse(is.na(occ\_wf\_1990), NA, match(occ\_wf\_1990, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_wf\_1990 = ifelse(is.na(occ\_wf\_1990), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_1990]),

occ\_hd\_1991 = ifelse(V19401 == 600, 580, V19401), occ\_hd\_1991 = na\_codes(occ\_hd\_1991, c(999, 0)),

occ.cw.position\_hd\_1991 = ifelse(is.na(occ\_hd\_1991), NA, match(occ\_hd\_1991, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_hd\_1991 = ifelse(is.na(occ\_hd\_1991 ), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_1991]),

occ\_wf\_1991 = ifelse(V19703 == 600, 580, V19703), occ\_wf\_1991 = na\_codes(occ\_wf\_1991, c(999, 0)),

occ.cw.position\_wf\_1991 = ifelse(is.na(occ\_wf\_1991), NA, match(occ\_wf\_1991, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_wf\_1991 = ifelse(is.na(occ\_wf\_1991), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_1991]),

occ\_hd\_1999 = ifelse(ER13215 == 600, 580, ER17226), occ\_hd\_1999 = na\_codes(occ\_hd\_1999, c(999, 810, 0)),

occ.cw.position\_hd\_1999 = ifelse(is.na(occ\_hd\_1999), NA, match(occ\_hd\_1999, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_hd\_1999 = ifelse(is.na(occ\_hd\_1999 ), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_1999]),

occ\_wf\_1999 = ifelse(ER13727 == 600, 580, ER17796), occ\_wf\_1999 = na\_codes(occ\_wf\_1999, c(999, 810, 0)),

occ.cw.position\_wf\_1999 = ifelse(is.na(occ\_wf\_1999), NA, match(occ\_wf\_1999, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_wf\_1999 = ifelse(is.na(occ\_wf\_1999 ), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_1999]),

occ\_hd\_2001 = ifelse(ER17226 == 600, 580, ER17226), occ\_hd\_2001 = na\_codes(occ\_hd\_2001, c(999, 810, 0)),

occ.cw.position\_hd\_2001 = ifelse(is.na(occ\_hd\_2001), NA, match(occ\_hd\_2001, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_hd\_2001 = ifelse(is.na(occ\_hd\_2001 ), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2001]),

occ\_wf\_2001 = ifelse(ER17796 == 600, 580, ER17796), occ\_wf\_2001 = na\_codes(occ\_wf\_2001, c(999, 810, 0)),

occ.cw.position\_wf\_2001 = ifelse(is.na(occ\_wf\_2001), NA, match(occ\_wf\_2001, crosswalk.occ$occ1970, nomatch = NA)),

occ2010\_wf\_2001 = ifelse(is.na(occ\_wf\_2001 ), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2001]),

# For 2009-2015, we create variables that recode the 2000 census occupation codes to 2010 occ codes w/ the IPUMS crosswalk

occ\_hd\_2009 = na\_codes(ER42167, c(0, 999)),

occ.cw.position\_hd\_2009 = ifelse(is.na(occ\_hd\_2009), NA, match(occ\_hd\_2009, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_hd\_2009 = ifelse(is.na(occ\_hd\_2009), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2009]),

occ\_wf\_2009 = na\_codes(ER42419, c(0, 999)),

occ.cw.position\_wf\_2009 = ifelse(is.na(occ\_wf\_2009), NA, match(occ\_wf\_2009, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_wf\_2009 = ifelse(is.na(occ\_wf\_2009), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2009]),

occ\_hd\_2011 = na\_codes(ER47479, c(0, 999)),

occ.cw.position\_hd\_2011 = ifelse(is.na(occ\_hd\_2011), NA, match(occ\_hd\_2011, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_hd\_2011 = ifelse(is.na(occ\_hd\_2011), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2011]),

occ\_wf\_2011 = na\_codes(ER47736, c(0, 999)),

occ.cw.position\_wf\_2011 = ifelse(is.na(occ\_wf\_2011), NA, match(occ\_wf\_2011, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_wf\_2011 = ifelse(is.na(occ\_wf\_2011), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2011]),

occ\_hd\_2015 = na\_codes(ER60194, c(0, 999)),

occ.cw.position\_hd\_2015 = ifelse(is.na(occ\_hd\_2015), NA, match(occ\_hd\_2015, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_hd\_2015 = ifelse(is.na(occ\_hd\_2015), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2015]),

occ\_wf\_2015 = na\_codes(ER60457, c(0, 999)),

occ.cw.position\_wf\_2015 = ifelse(is.na(occ\_wf\_2015), NA, match(occ\_wf\_2015, crosswalk.occ$occ2000, nomatch = NA)),

occ2010\_wf\_2015 = ifelse(is.na(occ\_wf\_2015), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2015]),

# For 2017 & 2019, we create variables that recode the ACS 2010 occupation codes to 2010 occ codes w/ the IPUMS crosswalk

occ\_hd\_2017 = na\_codes(ER66195, c(0, 999, 9999)),

occ.cw.position\_hd\_2017 = ifelse(is.na(occ\_hd\_2017), NA, match(occ\_hd\_2017, crosswalk.occ$acs2010, nomatch = NA)),

occ2010\_hd\_2017 = ifelse(is.na(occ\_hd\_2017), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2017]),

occ\_wf\_2017 = na\_codes(ER66470, c(0, 999, 9999)),

occ.cw.position\_wf\_2017 = ifelse(is.na(occ\_wf\_2017), NA, match(occ\_wf\_2017, crosswalk.occ$acs2010, nomatch = NA)),

occ2010\_wf\_2017 = ifelse(is.na(occ\_wf\_2017), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2017]),

occ\_hd\_2019 = na\_codes(ER72195, c(0, 999, 9999)),

occ.cw.position\_hd\_2019 = ifelse(is.na(occ\_hd\_2019), NA, match(occ\_hd\_2019, crosswalk.occ$acs2010, nomatch = NA)),

occ2010\_hd\_2019 = ifelse(is.na(occ\_hd\_2019), NA, crosswalk.occ$occ2010[occ.cw.position\_hd\_2019]),

occ\_wf\_2019 = na\_codes(ER72472, c(0, 999, 9999)),

occ.cw.position\_wf\_2019 = ifelse(is.na(occ\_wf\_2019), NA, match(occ\_wf\_2019, crosswalk.occ$acs2010, nomatch = NA)),

occ2010\_wf\_2019 = ifelse(is.na(occ\_wf\_2019), NA, crosswalk.occ$occ2010[occ.cw.position\_wf\_2019]),

# Industry variables- as with occupation, we use the non-recoded 1970 PSID industry values in 1980

# to generate a harmonized scheme of industry dummies (more details in the appendix). We also use the

# recoded industry values (1980 PSID industry values recoded to 1970 census industry) for those recoded by the PSID,

# which we use to merge to IPUMs- generated industry characteristics.

# First removing missing values

ind\_hd\_1980 = na\_codes(V7101, c(0, 99)), ind\_wf\_1980 = na\_codes(V7199, c(0,99)),

ind.recode\_hd\_1980 = na\_codes(V7101\_A, c(0, 99)), ind.recode\_wf\_1980 = na\_codes(V7199\_A, c(0,99)),

# Similar to occupation, we use a crosswalk to generate variables that recode the 1970 census industry codes

# to 1990 census industry codes

ind.recode.cw.position\_hd\_1980 = ifelse(is.na(ind.recode\_hd\_1980), NA, match(ind.recode\_hd\_1980, crosswalk.ind$ind1970, nomatch = NA)),

ind.recode.cw.position\_wf\_1980 = ifelse(is.na(ind.recode\_wf\_1980), NA, match(ind.recode\_wf\_1980, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_1980 = ifelse(is.na(ind.recode\_hd\_1980), NA, crosswalk.ind$ind1990[ind.recode.cw.position\_hd\_1980]),

ind1990\_wf\_1980 = ifelse(is.na(ind.recode\_wf\_1980), NA, crosswalk.ind$ind1990[ind.recode.cw.position\_wf\_1980]),

# For 1981-2001, we just do the crosswalk procedure, as in 1981-2001 industry values were coded according to the 1970 census scheme

ind\_hd\_1981 = na\_if(V7713, 999), ind\_wf\_1981 = na\_if(V7886, 999),

ind.cw.position\_hd\_1981 = ifelse(is.na(ind\_hd\_1981), NA, match(ind\_hd\_1981, crosswalk.ind$ind1970, nomatch = NA)),

ind.cw.position\_wf\_1981 = ifelse(is.na(ind\_wf\_1981), NA, match(ind\_wf\_1981, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_1981 = ifelse(is.na(ind\_hd\_1981), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_1981]),

ind1990\_wf\_1981 = ifelse(is.na(ind\_wf\_1981), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_1981]),

ind\_hd\_1990 = na\_if(V18102, 999), ind\_wf\_1990 = na\_if(V18404, 999),

ind.cw.position\_hd\_1990 = ifelse(is.na(ind\_hd\_1990), NA, match(ind\_hd\_1990, crosswalk.ind$ind1970, nomatch = NA)),

ind.cw.position\_wf\_1990 = ifelse(is.na(ind\_wf\_1990), NA, match(ind\_wf\_1990, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_1990 = ifelse(is.na(ind\_hd\_1990), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_1990]),

ind1990\_wf\_1990 = ifelse(is.na(ind\_wf\_1990), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_1990]),

ind\_hd\_1991 = na\_if(V19402, 999), ind\_wf\_1991 = na\_if(V19704, 999),

ind.cw.position\_hd\_1991 = ifelse(is.na(ind\_hd\_1991), NA, match(ind\_hd\_1991, crosswalk.ind$ind1970, nomatch = NA)),

ind.cw.position\_wf\_1991 = ifelse(is.na(ind\_wf\_1991), NA, match(ind\_wf\_1991, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_1991 = ifelse(is.na(ind\_hd\_1991), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_1991]),

ind1990\_wf\_1991 = ifelse(is.na(ind\_wf\_1991), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_1991]),

ind\_hd\_1999 = na\_if(ER13216, 999), ind\_wf\_1999 = na\_if(ER13728, 999),

ind.cw.position\_hd\_1999 = ifelse(is.na(ind\_hd\_1999), NA, match(ind\_hd\_1999, crosswalk.ind$ind1970, nomatch = NA)),

ind.cw.position\_wf\_1999 = ifelse(is.na(ind\_wf\_1999), NA, match(ind\_wf\_1999, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_1999 = ifelse(is.na(ind\_hd\_1999), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_1999]),

ind1990\_wf\_1999 = ifelse(is.na(ind\_wf\_1999), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_1999]),

ind\_hd\_2001 = na\_if(ER17227, 999), ind\_wf\_2001 = na\_if(ER17797, 999),

ind.cw.position\_hd\_2001 = ifelse(is.na(ind\_hd\_2001), NA, match(ind\_hd\_2001, crosswalk.ind$ind1970, nomatch = NA)),

ind.cw.position\_wf\_2001 = ifelse(is.na(ind\_wf\_2001), NA, match(ind\_wf\_2001, crosswalk.ind$ind1970, nomatch = NA)),

ind1990\_hd\_2001 = ifelse(is.na(ind\_hd\_2001), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_2001]),

ind1990\_wf\_2001 = ifelse(is.na(ind\_wf\_2001), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_2001]),

# For 2011-2015: two industry values in the PSID have no match in the crosswalk nor any matches in any

# codebooks we could find for Industry 2000, which are the industry codes used by the PSID in 2011-2015.

# One value is 120 (2 respondents, both male) and falls within the non-durable manufacturing range.

# We recode this value to 127, also in the non-durable manufacturing range.

# Another value is 775 (5 respondents, all female) and falls within the Management, Administrative and Support,

# and Waste Management Services range: we recode this value to 779, which is coded as sanitary services

ind\_hd\_2009 = na\_codes(ER42168, c(0, 999)), ind\_wf\_2009 = na\_codes(ER42420, c(0, 999)),

ind\_hd\_2009 = ifelse(ind\_hd\_2009 == 120, 127, ind\_hd\_2009), ind\_wf\_2009 = ifelse(ind\_wf\_2009 == 775, 779, ind\_wf\_2009),

ind.cw.position\_hd\_2009 = ifelse(is.na(ind\_hd\_2009), NA, match(ind\_hd\_2009, crosswalk.ind$ind2000, nomatch = NA)),

ind.cw.position\_wf\_2009 = ifelse(is.na(ind\_wf\_2009), NA, match(ind\_wf\_2009, crosswalk.ind$ind2000, nomatch = NA)),

ind1990\_hd\_2009 = ifelse(is.na(ind\_hd\_2009), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_2009]),

ind1990\_wf\_2009 = ifelse(is.na(ind\_wf\_2009), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_2009]),

ind\_hd\_2011 = na\_codes(ER47480, c(0, 999)), ind\_wf\_2011 = na\_codes(ER47737, c(0, 999)),

ind\_hd\_2011 = ifelse(ind\_hd\_2011 == 120, 127, ind\_hd\_2011), ind\_wf\_2011 = ifelse(ind\_wf\_2011 == 775, 779, ind\_wf\_2011),

ind.cw.position\_hd\_2011 = ifelse(is.na(ind\_hd\_2011), NA, match(ind\_hd\_2011, crosswalk.ind$ind2000, nomatch = NA)),

ind.cw.position\_wf\_2011 = ifelse(is.na(ind\_wf\_2011), NA, match(ind\_wf\_2011, crosswalk.ind$ind2000, nomatch = NA)),

ind1990\_hd\_2011 = ifelse(is.na(ind\_hd\_2011), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_2011]),

ind1990\_wf\_2011 = ifelse(is.na(ind\_wf\_2011), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_2011]),

ind\_hd\_2015 = na\_codes(ER60195, c(0, 999)), ind\_wf\_2015 = na\_codes(ER60458, c(0, 999)),

ind\_hd\_2015 = ifelse(ind\_hd\_2015 == 120, 127, ind\_hd\_2015), ind\_wf\_2015 = ifelse(ind\_wf\_2015 == 775, 779, ind\_wf\_2015),

ind.cw.position\_hd\_2015 = ifelse(is.na(ind\_hd\_2015), NA, match(ind\_hd\_2015, crosswalk.ind$ind2000, nomatch = NA)),

ind.cw.position\_wf\_2015 = ifelse(is.na(ind\_wf\_2015), NA, match(ind\_wf\_2015, crosswalk.ind$ind2000, nomatch = NA)),

ind1990\_hd\_2015 = ifelse(is.na(ind\_hd\_2015), NA, crosswalk.ind$ind1990[ind.cw.position\_hd\_2015]),

ind1990\_wf\_2015 = ifelse(is.na(ind\_wf\_2015), NA, crosswalk.ind$ind1990[ind.cw.position\_wf\_2015]),

ind\_hd\_2017 = na\_if(ER66196, 9999), ind\_wf\_2017 = na\_if(ER66471, 9999),

ind.cw.position\_hd\_2017 = ifelse(is.na(ind\_hd\_2017), NA, match(ind\_hd\_2017, crosswalk.ind.2012$ind2012, nomatch = NA)),

ind.cw.position\_wf\_2017 = ifelse(is.na(ind\_wf\_2017), NA, match(ind\_wf\_2017, crosswalk.ind.2012$ind2012, nomatch = NA)),

ind1990\_hd\_2017 = ifelse(is.na(ind\_hd\_2017), NA, crosswalk.ind.2012$ind1990[ind.cw.position\_hd\_2017]),

ind1990\_wf\_2017 = ifelse(is.na(ind\_wf\_2017), NA, crosswalk.ind.2012$ind1990[ind.cw.position\_wf\_2017]),

ind\_hd\_2019 = na\_if(ER72196, 9999), ind\_wf\_2019 = na\_if(ER72473, 9999),

ind.cw.position\_hd\_2019 = ifelse(is.na(ind\_hd\_2019), NA, match(ind\_hd\_2019, crosswalk.ind.2012$ind2012, nomatch = NA)),

ind.cw.position\_wf\_2019 = ifelse(is.na(ind\_wf\_2019), NA, match(ind\_wf\_2019, crosswalk.ind.2012$ind2012, nomatch = NA)),

ind1990\_hd\_2019 = ifelse(is.na(ind\_hd\_2019), NA, crosswalk.ind.2012$ind1990[ind.cw.position\_hd\_2019]),

ind1990\_wf\_2019 = ifelse(is.na(ind\_wf\_2019), NA, crosswalk.ind.2012$ind1990[ind.cw.position\_wf\_2019]),

# Unionization variables - 9 and 8 are implicitly coded as missing

# When more than one job is available, we use values for current main job

union\_hd\_1980 = case\_when(V7098 == 1 ~ 1, V7098 == 5 ~ 0), union\_wf\_1980 = case\_when(V7196 == 1 ~ 1, V7196 == 5 ~ 0),

union\_hd\_1981 = case\_when(V7709 == 1 ~ 1, V7709 == 5 ~ 0), union\_wf\_1981 = case\_when(V7882 == 1 ~ 1, V7882 == 5 ~ 0),

union\_hd\_1990 = case\_when(V18099 == 1 ~ 1, V18099 == 5 ~ 0), union\_wf\_1990 = case\_when(V18401 == 1 ~ 1, V18401 == 5 ~ 0),

union\_hd\_1991 = case\_when(V19399 == 1 ~ 1, V19399 == 5 ~ 0), union\_wf\_1991 = case\_when(V19701 == 1 ~ 1, V19701 == 5 ~ 0),

union\_hd\_1999 = case\_when(ER13213 == 1 ~ 1, ER13213 == 5 ~ 0), union\_wf\_1999 = case\_when(ER13725 == 1 ~ 1, ER13725 == 5 ~ 0),

union\_hd\_2001 = case\_when(ER17224 == 1 ~ 1, ER17224 == 5 ~ 0), union\_wf\_2001 = case\_when(ER17794 == 1 ~ 1, ER17794 == 5 ~ 0),

union\_hd\_2009 = case\_when(ER42178 == 1 ~ 1, ER42178 == 5 ~ 0), union\_wf\_2009 = case\_when(ER42430 == 1 ~ 1, ER42430 == 5 ~ 0),

union\_hd\_2011 = case\_when(ER47491 == 1 ~ 1, ER47491 == 5 ~ 0), union\_wf\_2011 = case\_when(ER47748 == 1 ~ 1, ER47748 == 5 ~ 0),

union\_hd\_2015 = case\_when(ER60206 == 1 ~ 1, ER60206 == 5 ~ 0), union\_wf\_2015 = case\_when(ER60469 == 1 ~ 1, ER60469 == 5 ~ 0),

union\_hd\_2017 = case\_when(ER66207 == 1 ~ 1, ER66207 == 5 ~ 0), union\_wf\_2017 = case\_when(ER66482 == 1 ~ 1, ER66482 == 5 ~ 0),

union\_hd\_2019 = case\_when(ER72207 == 1 ~ 1, ER72207 == 5 ~ 0), union\_wf\_2019 = case\_when(ER72484 == 1 ~ 1, ER72484 == 5 ~ 0),

# Self-employment variables: in early years, we use single-job reports because that’s all that is available (?). For later years, we use all jobs,

# and R is coded as self-employed if they report being self-employed in any job, following Blau & Kahn

self.emp\_alljobs\_hd\_1980 = case\_when(V7096 %in% c(2, 3) ~ 1, V7096 == 1 ~ 0), self.emp\_alljobs\_wf\_1980 = case\_when(V7194 %in% c(2, 3) ~ 1, V7194 == 1 ~ 0),

self.emp\_alljobs\_hd\_1981 = case\_when(V7707 %in% c(2, 3) ~ 1, V7707 == 1 ~ 0), self.emp\_alljobs\_wf\_1981 = case\_when(V7880 %in% c(2, 3) ~ 1, V7880 == 1 ~ 0),

self.emp\_alljobs\_hd\_1990 = case\_when(V18096 %in% c(2, 3) ~ 1, V18096 == 1 ~ 0), self.emp\_alljobs\_wf\_1991 = case\_when(V18398 %in% c(2, 3) ~ 1, V18398 == 1 ~ 0),

self.emp\_alljobs\_hd\_1991 = case\_when(V19396 %in% c(2, 3) ~ 1, V19396 == 1 ~ 0), self.emp\_alljobs\_wf\_1991 = case\_when(V19698 %in% c(2, 3) ~ 1, V19698 == 1 ~ 0),

self.emp\_alljobs\_wf\_1999 = case\_when(ER13292 %in% c(2, 3) ~ 1, ER13292 == 1 ~ 0), self.emp\_alljobs\_wf\_1999 = case\_when(ER13722 %in% c(2, 3) ~ 1, ER13722 == 1 ~ 0),

self.emp\_alljobs\_hd\_2001 = case\_when(ER17303 %in% c(2, 3) ~ 1, ER17303 == 1 ~ 0), self.emp\_alljobs\_wf\_2001 = case\_when(ER17791 %in% c(2, 3) ~ 1, ER17791 == 1 ~ 0),

self.emp\_alljobs\_hd\_2009 = case\_when(ER42169 %in% c(2,3) | ER42230 %in% c(2,3) | ER42260 %in% c(2,3) | ER42290 %in% c(2,3) ~ 1,

ER42169 == 1 | ER42230 == 1 | ER42260 == 1 | ER42290 == 1 ~ 0),

self.emp\_alljobs\_wf\_2009 = case\_when(ER42421 %in% c(2,3) | ER42482 %in% c(2,3) | ER42512 %in% c(2,3) | ER42542 %in% c(2,3) ~ 1,

ER42421 == 1 | ER42482 == 1 | ER42512 == 1 | ER42542 == 1 ~ 0),

self.emp\_alljobs\_hd\_2011 = case\_when(ER47482 %in% c(2,3) | ER47543 %in% c(2,3) | ER47573 %in% c(2,3) | ER47603 %in% c(2,3) ~ 1,

ER47482 == 1 | ER47543 == 1 | ER47573 == 1 | ER47603 == 1 ~ 0),

self.emp\_alljobs\_wf\_2011 = case\_when(ER47739 %in% c(2,3) | ER47800 %in% c(2,3) | ER47830 %in% c(2,3) | ER47860 %in% c(2,3) ~ 1,

ER47739 == 1 | ER47800 == 1 | ER47830 == 1 | ER47860 == 1 ~ 0),

self.emp\_alljobs\_hd\_2015 = case\_when(ER60197 %in% c(2,3) | ER60258 %in% c(2,3) | ER60288 %in% c(2,3) | ER60318 %in% c(2,3) ~ 1,

ER60197 == 1 | ER60258 == 1 | ER60288 == 1 | ER60318 == 1 ~ 0),

self.emp\_alljobs\_wf\_2015 = case\_when(ER60460 %in% c(2,3) | ER60521 %in% c(2,3) | ER60551 %in% c(2,3) | ER60581 %in% c(2,3) ~ 1,

ER60460 == 1 | ER60521 == 1 | ER60551 == 1 | ER60581 == 1 ~ 0),

self.emp\_alljobs\_hd\_2017 = case\_when(ER66198 %in% c(2,3) | ER66261 %in% c(2,3) | ER66291 %in% c(2,3) | ER66321 %in% c(2,3) ~ 1,

ER66198 == 1 | ER66261 == 1 | ER66291 == 1 | ER66321 == 1 ~ 0),

self.emp\_alljobs\_wf\_2017 = case\_when(ER66473 %in% c(2,3) | ER66536 %in% c(2,3) | ER66566 %in% c(2,3) | ER66596 %in% c(2,3) ~ 1,

ER66473 == 1 | ER66536 == 1 | ER66566 == 1 | ER66596 == 1 ~ 0),

self.emp\_alljobs\_hd\_2019 = case\_when(ER72198 %in% c(2,3) | ER72263 %in% c(2,3) | ER72293 %in% c(2,3) | ER72321 %in% c(2,3) ~ 1,

ER72198 == 1 | ER72263 == 1 | ER72293 == 1 | ER72321 == 1 ~ 0),

self.emp\_alljobs\_wf\_2019 = case\_when(ER72475 %in% c(2,3) | ER72540 %in% c(2,3) | ER72568 %in% c(2,3) | ER72598 %in% c(2,3) ~ 1,

ER72475 == 1 | ER72540 == 1 | ER72568 == 1 | ER72598 == 1 ~ 0),

self.emp\_hd\_1980 = case\_when(V7096 %in% c(2, 3) ~ 1, V7096 == 1 ~ 0), self.emp\_wf\_1980 = case\_when(V7194 %in% c(2, 3) ~ 1, V7194 == 1 ~ 0),

self.emp\_hd\_1981 = case\_when(V7707 %in% c(2, 3) ~ 1, V7707 == 1 ~ 0), self.emp\_wf\_1981 = case\_when(V7880 %in% c(2, 3) ~ 1, V7880 == 1 ~ 0),

self.emp\_hd\_1990 = case\_when(V18096 %in% c(2, 3) ~ 1, V18096 == 1 ~ 0), self.emp\_wf\_1991 = case\_when(V18398 %in% c(2, 3) ~ 1, V18398 == 1 ~ 0),

self.emp\_hd\_1991 = case\_when(V19396 %in% c(2, 3) ~ 1, V19396 == 1 ~ 0), self.emp\_wf\_1991 = case\_when(V19698 %in% c(2, 3) ~ 1, V19698 == 1 ~ 0),

self.emp\_wf\_1999 = case\_when(ER13292 %in% c(2, 3) ~ 1, ER13292 == 1 ~ 0), self.emp\_wf\_1999 = case\_when(ER13722 %in% c(2, 3) ~ 1, ER13722 == 1 ~ 0),

self.emp\_hd\_2001 = case\_when(ER17303 %in% c(2, 3) ~ 1, ER17303 == 1 ~ 0), self.emp\_wf\_2001 = case\_when(ER17791 %in% c(2, 3) ~ 1, ER17791 == 1 ~ 0),

self.emp\_hd\_2009 = case\_when(ER42169 %in% c(2,3) ~ 1, ER42169 == 1 ~ 0), self.emp\_wf\_2009 = case\_when(ER42421 %in% c(2,3) ~ 1, ER42421 == 1 ~ 0),

self.emp\_hd\_2011 = case\_when(ER47482 %in% c(2,3) ~ 1, ER47482 == 1 ~ 0), self.emp\_wf\_2011 = case\_when(ER47739 %in% c(2,3) ~ 1, ER47739 == 1 ~ 0),

self.emp\_hd\_2015 = case\_when(ER60197 %in% c(2,3) ~ 1, ER60197 == 1 ~ 0), self.emp\_wf\_2015 = case\_when(ER60460 %in% c(2,3) ~ 1, ER60460 == 1 ~ 0),

self.emp\_hd\_2017 = case\_when(ER66198 %in% c(2,3) ~ 1, ER66198 == 1 ~ 0), self.emp\_wf\_2017 = case\_when(ER66473 %in% c(2,3) ~ 1, ER66473 == 1 ~ 0),

self.emp\_hd\_2019 = case\_when(ER72198 %in% c(2,3) ~ 1, ER72198 == 1 ~ 0), self.emp\_wf\_2019 = case\_when(ER72475 %in% c(2,3) ~ 1, ER72475 == 1 ~ 0),

# Government Job: similar to self-employed, we use single-job reports for early years, multiple reports in later years

govt.job\_alljobs\_hd\_1980 = case\_when(V7097 == 1 ~ 1, V7097 == 5 ~ 0), govt.job\_alljobs\_wf\_1980 = case\_when(V7195 == 1 ~ 1, V7195 == 5 ~ 0),

govt.job\_alljobs\_hd\_1981 = case\_when(V7708 == 1 ~ 1, V7708 == 5 ~ 0), govt.job\_alljobs\_wf\_1981 = case\_when(V7881 == 1 ~ 1, V7881 == 5 ~ 0),

govt.job\_alljobs\_hd\_1990 = case\_when(V18098 %in% c(1,2,3) ~ 1, V18098 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_1990 = case\_when(V18400 %in% c(1,2,3) ~ 1, V18400 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_1991 = case\_when(V19398 %in% c(1,2,3) ~ 1, V19398 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_1991 = case\_when(V19700 %in% c(1,2,3) ~ 1, V19700 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_1999 = case\_when(ER13212 %in% c(1,2,3) | ER13294 %in% c(1,2,3) ~ 1,

ER13212 %in% c(4,7) | ER13294 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_1999 = case\_when(ER13724 %in% c(1,2,3) | ER13806 %in% c(1,2,3) ~ 1,

ER13724 %in% c(4,7) | ER13806 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2001 = case\_when(ER17223 %in% c(1,2,3) | ER17305 %in% c(1,2,3) ~ 1,

ER17223 %in% c(4,7) | ER17305 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2001 = case\_when(ER17793 %in% c(1,2,3) | ER17875 %in% c(1,2,3) ~ 1,

ER17793 %in% c(4,7) | ER17875 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2009 = case\_when(ER42171 %in% c(1,2,3) | ER42232 %in% c(1,2,3) | ER42262 %in% c(1,2,3) | ER42292 %in% c(1,2,3) ~ 1,

ER42171 %in% c(4,7) | ER42232 %in% c(4,7) | ER42262 %in% c(4,7) | ER42292 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2009 = case\_when(ER42423 %in% c(1,2,3) | ER42484 %in% c(1,2,3) | ER42514 %in% c(1,2,3) | ER42544 %in% c(1,2,3) ~ 1,

ER42423 %in% c(4,7) | ER42484 %in% c(4,7) | ER42514 %in% c(4,7) | ER42544 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2011 = case\_when(ER47484 %in% c(1,2,3) | ER47545 %in% c(1,2,3) | ER47575 %in% c(1,2,3) | ER47605 %in% c(1,2,3) ~ 1,

ER47484 %in% c(4,7) | ER47545 %in% c(4,7) | ER47575 %in% c(4,7) | ER47605 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2011 = case\_when(ER47741 %in% c(1,2,3) | ER47802 %in% c(1,2,3) | ER47832 %in% c(1,2,3) | ER47862 %in% c(1,2,3) ~ 1,

ER47741 %in% c(4,7) | ER47802 %in% c(4,7) | ER47832 %in% c(4,7) | ER47862 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2015 = case\_when(ER60199 %in% c(1,2,3) | ER60260 %in% c(1,2,3) | ER60290 %in% c(1,2,3) | ER60320 %in% c(1,2,3) ~ 1,

ER60199 %in% c(4,7) | ER60260 %in% c(4,7) | ER60290 %in% c(4,7) | ER60320 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2015 = case\_when(ER60462 %in% c(1,2,3) | ER60523 %in% c(1,2,3) | ER60553 %in% c(1,2,3) | ER60583 %in% c(1,2,3) ~ 1,

ER60462 %in% c(4,7) | ER60523 %in% c(4,7) | ER60553 %in% c(4,7) | ER60583 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2017 = case\_when(ER66200 %in% c(1,2,3) | ER66263 %in% c(1,2,3) | ER66293 %in% c(1,2,3) | ER66323 %in% c(1,2,3) ~ 1,

ER66200 %in% c(4,7) | ER66263 %in% c(4,7) | ER66293 %in% c(4,7) | ER66323 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2017 = case\_when(ER66475 %in% c(1,2,3) | ER66538 %in% c(1,2,3) | ER66568 %in% c(1,2,3) | ER66598 %in% c(1,2,3) ~ 1,

ER66475 %in% c(4,7) | ER66538 %in% c(4,7) | ER66568 %in% c(4,7) | ER66598 %in% c(4,7) ~ 0),

govt.job\_alljobs\_hd\_2019 = case\_when(ER72200 %in% c(1,2,3) | ER72263 %in% c(1,2,3) | ER72293 %in% c(1,2,3) | ER72323 %in% c(1,2,3) ~ 1,

ER72200 %in% c(4,7) | ER72263 %in% c(4,7) | ER72293 %in% c(4,7) | ER72323 %in% c(4,7) ~ 0),

govt.job\_alljobs\_wf\_2019 = case\_when(ER72477 %in% c(1,2,3) | ER72540 %in% c(1,2,3) | ER72570 %in% c(1,2,3) | ER72600 %in% c(1,2,3) ~ 1,

ER72477 %in% c(4,7) | ER72540 %in% c(4,7) | ER72570 %in% c(4,7) | ER72600 %in% c(4,7) ~ 0),

# Government Job: Using main job reports only

govt.job\_hd\_1980 = case\_when(V7097 == 1 ~ 1, V7097 == 5 ~ 0), govt.job\_wf\_1980 = case\_when(V7195 == 1 ~ 1, V7195 == 5 ~ 0),

govt.job\_hd\_1981 = case\_when(V7708 == 1 ~ 1, V7708 == 5 ~ 0), govt.job\_wf\_1981 = case\_when(V7881 == 1 ~ 1, V7881 == 5 ~ 0),

govt.job\_hd\_1990 = case\_when(V18098 %in% c(1,2,3) ~ 1, V18098 %in% c(4,7) ~ 0),

govt.job\_wf\_1990 = case\_when(V18400 %in% c(1,2,3) ~ 1, V18400 %in% c(4,7) ~ 0),

govt.job\_hd\_1991 = case\_when(V19398 %in% c(1,2,3) ~ 1, V19398 %in% c(4,7) ~ 0),

govt.job\_wf\_1991 = case\_when(V19700 %in% c(1,2,3) ~ 1, V19700 %in% c(4,7) ~ 0),

govt.job\_hd\_1999 = case\_when(ER13212 %in% c(1,2,3) | ER13294 %in% c(1,2,3) ~ 1,

ER13212 %in% c(4,7) | ER13294 %in% c(4,7) ~ 0),

govt.job\_wf\_1999 = case\_when(ER13724 %in% c(1,2,3) | ER13806 %in% c(1,2,3) ~ 1,

ER13724 %in% c(4,7) | ER13806 %in% c(4,7) ~ 0),

govt.job\_hd\_2001 = case\_when(ER17223 %in% c(1,2,3) | ER17305 %in% c(1,2,3) ~ 1,

ER17223 %in% c(4,7) | ER17305 %in% c(4,7) ~ 0),

govt.job\_wf\_2001 = case\_when(ER17793 %in% c(1,2,3) | ER17875 %in% c(1,2,3) ~ 1,

ER17793 %in% c(4,7) | ER17875 %in% c(4,7) ~ 0),

govt.job\_hd\_2009 = case\_when(ER42171 %in% c(1,2,3) ~ 1, ER42171 %in% c(4,7) ~ 0),

govt.job\_wf\_2009 = case\_when(ER42423 %in% c(1,2,3) ~ 1, ER42423 %in% c(4,7) ~ 0),

govt.job\_hd\_2011 = case\_when(ER47484 %in% c(1,2,3) ~ 1, ER47484 %in% c(4,7) ~ 0),

govt.job\_wf\_2011 = case\_when(ER47741 %in% c(1,2,3) ~ 1, ER47741 %in% c(4,7) ~ 0),

govt.job\_hd\_2015 = case\_when(ER60199 %in% c(1,2,3) ~ 1, ER60199 %in% c(4,7) ~ 0),

govt.job\_wf\_2015 = case\_when(ER60462 %in% c(1,2,3) ~ 1, ER60462 %in% c(4,7) ~ 0),

govt.job\_hd\_2017 = case\_when(ER66200 %in% c(1,2,3) ~ 1, ER66200 %in% c(4,7) ~ 0),

govt.job\_wf\_2017 = case\_when(ER66475 %in% c(1,2,3) ~ 1, ER66475 %in% c(4,7) ~ 0),

govt.job\_hd\_2019 = case\_when(ER72200 %in% c(1,2,3) ~ 1, ER72200 %in% c(4,7) ~ 0),

govt.job\_wf\_2019 = case\_when(ER72477 %in% c(1,2,3) ~ 1, ER72477 %in% c(4,7) ~ 0),

# Wage data: collected for wage income earned in prior year. For head and wife, using total labor income for 1981

# as PSID doesn't collect separate values for wage/salary income and self employment/farm income for wives in

# early years. Editing topcode for 1981 following Blau & Kahn.

# Adjusting dollars to 2010 dollars using consumer price index

# by multiplying income year to 1999 dollars, then 1999 dollars to 2010 values (1.309)

# https://cps.ipums.org/cps/cpi99.shtml

wages\_hd\_1980 = NA, wages\_wf\_1980 = NA,

wages\_hd\_1981 = ifelse(V8066 == 99999, 99999\*1.45, V8066) \* 2.022 \* 1.309,

wages\_wf\_1981 = ifelse(V7580 == 99999, 99999\*1.45, V7580) \* 2.022 \* 1.309,

wages\_hd\_1990 = V18878 \* 1.344 \* 1.309,

wages\_wf\_1990 = V17836 \* 1.344 \* 1.309,

wages\_hd\_1991 = V20178 \* 1.275 \* 1.309,

wages\_wf\_1991 = V19136 \* 1.275 \* 1.309,

wages\_hd\_1999 = NA, wages\_wf\_1999 = NA,

wages\_hd\_2001 = ER20443 \* 0.967 \* 1.309,

wages\_wf\_2001 = ER20447 \* 0.967 \* 1.309,

wages\_hd\_2009 = NA, wages\_wf\_2009 = NA,

wages\_hd\_2011 = ER52237 \* 0.764 \* 1.309,

wages\_wf\_2011 = ER52249 \* 0.764 \* 1.309,

wages\_hd\_2015 = NA, wages\_wf\_2015 = NA,

wages\_hd\_2017 = ER71293 \* 0.694 \* 1.309, wages\_wf\_2017 = ER71321 \* 0.694 \* 1.309,

wages\_hd\_2019 = ER77315 \* 0.663 \* 1.309, wages\_wf\_2019 = ER77343 \* 0.663 \* 1.309,

faminc\_1980 = V7412 \* 2.295 \* 1.309,

faminc\_1981 = V8065 \* 2.022 \* 1.309,

faminc\_1990 = V18875 \* 1.344 \* 1.309,

faminc\_1991 = V20175 \* 1.275 \* 1.309,

faminc\_1999 = ER16462\* 1.022 \* 1.309,

faminc\_2001 = ER20456 \* 0.967 \* 1.309,

faminc\_2009 = ER46935 \* 0.774 \* 1.309,

faminc\_2011 = ER52343 \* 0.764 \* 1.309,

faminc\_2015 = ER65349 \* 0.704 \* 1.309,

faminc\_2017 = ER71426 \* 0.694 \* 1.309,

faminc\_2019 = ER77448 \* 0.663 \* 1.309,

# Employer Tenure: not available in 1980, but collected in 1981

# for the later years, we use months and years with this employer to generate an equivalent measure to the early yrs

emp.tenure\_hd\_1980 = NA, emp.tenure\_wf\_1980 = NA,

emp.tenure\_hd\_1981 = ifelse(V7711 == 999, NA, V7711), emp.tenure\_wf\_1981 = ifelse(V7884 == 999, NA, V7884),

emp.tenure\_hd\_1990 = ifelse(V18120 == 999, NA, V18120), emp.tenure\_wf\_1990 = ifelse(V18422 == 999, NA, V18422),

emp.tenure\_hd\_1991 = ifelse(V19420 == 999, NA, V19420), emp.tenure\_wf\_1991 = ifelse(V19722 == 999, NA, V19722),

emp.tenure.mon\_hd\_1999 = ifelse(ER13244 >= 98, NA, ER13244), emp.tenure.yr\_hd\_1999 = ifelse(ER13243 >= 98, NA, ER13243) \* 12,

emp.tenure.mon\_wf\_1999 = ifelse(ER13756 >= 98, NA, ER13756), emp.tenure.yr\_wf\_1999 = ifelse(ER13755 >= 98, NA, ER13755) \* 12,

emp.tenure.mon\_hd\_2001 = ifelse(ER17255 >= 98, NA, ER17255), emp.tenure.yr\_hd\_2001 = ifelse(ER17254 >= 98, NA, ER17254) \* 12,

emp.tenure.mon\_wf\_2001 = ifelse(ER17825 >= 98, NA, ER17825), emp.tenure.yr\_wf\_2001 = ifelse(ER17824 >= 98, NA, ER17824) \* 12,

emp.tenure.mon\_hd\_2009 = ifelse(ER42201 >= 98, NA, ER42201), emp.tenure.yr\_hd\_2009 = ifelse(ER42200 >= 98, NA, ER42200) \* 12,

emp.tenure.mon\_wf\_2009 = ifelse(ER42453 >= 98, NA, ER42453), emp.tenure.yr\_wf\_2009 = ifelse(ER42452 >= 98, NA, ER42452) \* 12,

emp.tenure.mon\_hd\_2011 = ifelse(ER47514 >= 98, NA, ER47514), emp.tenure.yr\_hd\_2011 = ifelse(ER47513 >= 98, NA, ER47513) \* 12,

emp.tenure.mon\_wf\_2011 = ifelse(ER47771 >= 98, NA, ER47771), emp.tenure.yr\_wf\_2011 = ifelse(ER47770 >= 98, NA, ER47770) \* 12,

emp.tenure.mon\_hd\_2015 = ifelse(ER60229 >= 98, NA, ER60229), emp.tenure.yr\_hd\_2015 = ifelse(ER60228 >= 98, NA, ER60228) \* 12,

emp.tenure.mon\_wf\_2015 = ifelse(ER60492 >= 98, NA, ER60492), emp.tenure.yr\_wf\_2015 = ifelse(ER60491 >= 98, NA, ER60491) \* 12,

emp.tenure.mon\_hd\_2017 = ifelse(ER66232 >= 98, NA, ER66232), emp.tenure.yr\_hd\_2017 = ifelse(ER66231 >= 98, NA, ER66231) \* 12,

emp.tenure.mon\_wf\_2017 = ifelse(ER66507 >= 98, NA, ER66507), emp.tenure.yr\_wf\_2017 = ifelse(ER66506 >= 98, NA, ER66506) \* 12,

emp.tenure.mon\_hd\_2019 = ifelse(ER72232 >= 98, NA, ER72232), emp.tenure.yr\_hd\_2019 = ifelse(ER72231 >= 98, NA, ER72231) \* 12,

emp.tenure.mon\_wf\_2019 = ifelse(ER72509 >= 98, NA, ER72509), emp.tenure.yr\_wf\_2019 = ifelse(ER72508 >= 98, NA, ER72508) \* 12,

# Age of youngest child in the household

age.youngest\_1980 = V7071,

age.youngest\_1981 = V7662,

age.youngest\_1990 = V18053,

age.youngest\_1991 = V19353,

age.youngest\_1999 = ER13014,

age.youngest\_2001 = ER17017,

age.youngest\_2009 = ER42021,

age.youngest\_2011 = ER47321,

age.youngest\_2015 = ER60022,

age.youngest\_2017 = ER66022,

age.youngest\_2019 = ER72022,

# Creating an indicator variable for whether the respondent is interviewed as a head/wife

# that meets our desired age range in the outcome years

samp.inc.1981 = case\_when(int.num\_1981 > 0 & seq.num\_1981 <= 20 & seq.num\_1981 != 0

& rel.head\_1981 %in% c("head", "wife") & age\_1981 %in% seq(30, 60, 1) ~ 1, TRUE ~ 0),

samp.inc.1991 = case\_when(int.num\_1991 > 0 & seq.num\_1991 <= 20 & seq.num\_1991 != 0

& rel.head\_1991 %in% c("head", "wife") & age\_1991 %in% seq(30, 60, 1) ~ 1, TRUE ~ 0),

samp.inc.2001 = case\_when(int.num\_2001 > 0 & seq.num\_2001 <= 20 & seq.num\_2001 != 0

& rel.head\_2001 %in% c("head", "wife") & age\_2001 %in% seq(30, 60, 1)~ 1, TRUE ~ 0),

samp.inc.2011 = case\_when(int.num\_2011 > 0 & seq.num\_2011 <= 20 & seq.num\_2011 != 0

& rel.head\_2011 %in% c("head", "wife") & age\_2011 %in% seq(30, 60, 1) ~ 1, TRUE ~ 0),

samp.inc.2017 = case\_when(int.num\_2017 > 0 & seq.num\_2017 <= 20 & seq.num\_2017 != 0

& rel.head\_2017 %in% c("head", "wife") & age\_2017 %in% seq(30, 60, 1)~ 1, TRUE ~ 0),

samp.inc.2019 = case\_when(int.num\_2019 > 0 & seq.num\_2019 <= 20 & seq.num\_2019 != 0

& rel.head\_2019 %in% c("head", "wife") & age\_2019 %in% seq(30, 60, 1)~ 1, TRUE ~ 0)) %>%

# Turning to long format where key = varname for all vars except time-constant variables

# (Sex, 1968 household and person numbers and individual identifier)

gather(key, value, -c(family\_id, person\_number, indiv.id,

samp\_error\_stratum, samp\_error\_cluster, female,

year.firstbornchild, samp.inc.1981, samp.inc.1991,

samp.inc.2001, samp.inc.2011, samp.inc.2017, samp.inc.2019)) %>%

# Creating year variable based on the covariate label

mutate(

year = case\_when(grepl("\_1980", key) ~ 1980, grepl("\_1981", key) ~ 1981,

grepl("\_1990", key) ~ 1990, grepl("\_1991", key) ~ 1991,

grepl("\_1999", key) ~ 1999, grepl("\_2001", key) ~ 2001,

grepl("\_2009", key) ~ 2009, grepl("\_2011", key) ~ 2011,

grepl("\_2015", key) ~ 2015, grepl("\_2017", key) ~ 2017,

grepl("\_2019", key) ~ 2019),

var = str\_remove(key, "\_[0-9]+[0-9]+"),

# Creating indicator variables for whether the respondent belongs in one of the immigrant or latino samples

# (families in the Latino sample are not included in main our target years)

imm.sample.97 = ifelse(family\_id >= 3001 & family\_id <= 3511, 1, 0),

imm.sample.17 = ifelse(family\_id >= 4001 & family\_id <= 4462, 1, 0),

latino.sample = ifelse(family\_id >= 7001 & family\_id <= 9308, 1, 0),

imm.sample = ifelse(imm.sample.97 == 1 | imm.sample.17 == 1, 1, 0)) %>%

dplyr::select(-key) %>%

# Grouping by individual id

group\_by(indiv.id) %>%

# Turning data back to wide format, each record is a person-year

spread(var, value, convert = T) %>%

mutate(

# Creating an indicator variable for whether the respondent was a non-interview or not a

# head/wife in that year, meaning the PSID will not have collected covariate data for R in that year

missing.interview = case\_when(int.num == 0 | seq.num > 20 | seq.num == 0 |

rel.head %!in% c("head", "wife") ~ 1, TRUE ~ 0),

# Creating indicator variable labeling whether respondent is head or wife in that year

hd.wife = ifelse(rel.head %in% c("head", "wife"), 1, 0))

# Creating a plot that identifies the proportion of respondents aged 25-65 in

# each year who are PSID heads or wives

prop.hdwife.plot <- psid\_raw %>%

filter(age >= 20 & age <= 65) %>%

filter(year %in% c(1981, 2019)) %>%

group\_by(year, age) %>%

summarise(wtd.pct = wpct(hd.wife, weight = perwt)[2] \* 100) %>%

ggplot(aes(x = age, y = wtd.pct)) +

geom\_line() +

facet\_wrap(~year) +

theme\_bw() +

labs(title = "Percent of PSID Sample Members Classified as Heads/Wives, by year and age",

y = "Percent of Sample Members Head/Wife (weighted)",

x = "Age") +

theme\_bw() +

theme(plot.title = element\_text(hjust = 0.5, face = "bold", size = 9),

axis.text.x = element\_text(angle = 45, hjust = 1))

# Creating a version of the data where we limit observations to individuals who are observed

# as heads or wives in our outcome years

psid\_clean <- psid\_raw %>%

# This keeps observations in 2015 and 2017 for individuals observed as heads/wives in our age range in 2017, and

# observations in 1980 and 1981 for individuals observed as heads/wives in our age range in 1981

filter(year %in% c(1980,1981) & samp.inc.1981 == 1 |

year %in% c(1990,1991) & samp.inc.1991 == 1 |

year %in% c(1999,2001) & samp.inc.2001 == 1 |

year %in% c(2009,2011) & samp.inc.2011 == 1 |

year %in% c(2017,2019) & samp.inc.2019 == 1) %>%

# Then we generate variables that take the value of the head for R's that are heads, value of spouse for spouses

mutate(avdeg = ifelse(rel.head == "head", advdeg\_hd, advdeg\_wf),

ann.wrk.hrs = ifelse(rel.head == "head", ann.wrk.hrs\_hd, ann.wrk.hrs\_wf),

ba = ifelse(rel.head == "head", ba\_hd, ba\_wf),

govt.job = ifelse(rel.head == "head", govt.job\_hd, govt.job\_wf),

govt.job\_alljobs = ifelse(rel.head == "head", govt.job\_alljobs\_hd, govt.job\_alljobs\_wf),

housework = ifelse(rel.head == "head", housework\_hd, housework\_wf),

ind.orig = ifelse(rel.head == "head", ind\_hd, ind\_wf),

occ.orig = ifelse(rel.head == "head", occ\_hd, occ\_wf),

# Note: for 1980, ind1990 and occ2010 will be NA for a subset of Rs who weren't recoded

# to the 1970 census scheme in the PSID

ind1990 = ifelse(rel.head == "head", ind1990\_hd, ind1990\_wf),

occ2010 = ifelse(rel.head == "head", occ2010\_hd, occ2010\_wf),

self.emp = ifelse(rel.head == "head", self.emp\_hd, self.emp\_wf),

self.emp\_alljobs = ifelse(rel.head == "head", self.emp\_alljobs\_hd, self.emp\_alljobs\_wf),

union = ifelse(rel.head == "head", union\_hd, union\_wf),

wages = ifelse(rel.head == "head", wages\_hd, wages\_wf),

wkswrk = ifelse(rel.head == "head", wks.wrk\_hd, wks.wrk\_wf),

yrs.ed.fam = ifelse(rel.head == "head", yrs.ed.fam\_hd, yrs.ed.fam\_wf),

emp.tenure.mon = ifelse(rel.head == "head", emp.tenure.mon\_hd, emp.tenure.mon\_wf),

emp.tenure.yr = ifelse(rel.head == "head", emp.tenure.yr\_hd, emp.tenure.yr\_wf),

emp.tenure = ifelse(rel.head == "head", emp.tenure\_hd, emp.tenure\_wf),

# Recoding remaining individual-level variables

age.first.birth = ifelse(!is.na(yr.born), year.firstbornchild - yr.born, year.firstbornchild - (year-age)),

marstat.hd = case\_when(marstat.hd == 1 ~ "married",

marstat.hd == 2 ~ "unmarried",

marstat.hd %in% 3:5 ~ "prev.married"),

age.youngest = na\_if(age.youngest, 999),

# Employer tenure is completely missing in 1980.

# In the later years some cases have missing year and 0 months: these are all coded to NA

emp.tenure = ifelse(year %in% c(1981, 1990, 1991), emp.tenure, ifelse( # For 1981 & 1991, take the months value

complete.cases(emp.tenure.yr), emp.tenure.yr + emp.tenure.mon, ifelse( # Add months & years (in months)

is.na(emp.tenure.yr) & complete.cases(emp.tenure.mon), emp.tenure.mon, NA))), # Cases w/no years, use months

hrs.wrk.wk = ifelse(is.na(wkswrk), NA, ifelse(wkswrk > 0, ann.wrk.hrs/wkswrk, 0)),

occ.orig = str\_pad(occ.orig, 3, pad = "0"),

# Industry 1990: "New" Mining Ind recorded in crosswalk, we recode to equivalent 1990 mining industry

ind1990 = ifelse(ind1990 == "New", "40", ind1990),

ind1990 = as.numeric(str\_pad(ind1990, 3, pad = "0"))) %>%

# Selecting variables at the individual-person level (all vars apply to R- if R is head, var takes value for head)

dplyr::select(family\_id, person\_number, samp\_error\_stratum, samp\_error\_cluster,

indiv.id, female, year, samp.inc.1981, samp.inc.2017, samp.inc.1991, samp.inc.2001, samp.inc.2011, latino.sample,

imm.sample.97, imm.sample.17, imm.sample, missing.interview, perwt, perwt.imm, perwt.lat, perwt.lat.imm, age, rel.head, marstat.hd, numkids.fu,

racehd, region, avdeg, ann.wrk.hrs, ba, govt.job, govt.job\_alljobs, housework, ind.orig, ind1990,

occ.orig, occ2010, self.emp, self.emp\_alljobs, union, wages, wkswrk, yrs.ed.fam, emp.tenure,

age.first.birth, hrs.wrk.wk, age.youngest, faminc, hd.wife) %>%

arrange(indiv.id, year) %>% # Arrange by year and individual id to create lead terms

group\_by(indiv.id) %>% # Group by id to create lead terms

# Creating lead terms which will be used to create sample selection variables

mutate(self.emp.lead = lead(self.emp), # Records self-employed status in the subsequent year

# Creates an indicator variable: if self-employed unobserved in covariate year, takes on the value in the subsequent year

self.emp.synth = ifelse(is.na(self.emp) & complete.cases(self.emp.lead), self.emp.lead, self.emp),

lead.ind1990 = lead(ind1990), lead.occ2010 = lead(occ2010), # Records industry and occ in the subsequent year

lead.ann.wrk.hrs = lead(ann.wrk.hrs), # Records annual work hours in the subsequent year

lead.wkswrk = lead(wkswrk), # Records weeks worked in the subsequent year

lead.hrs.wrk.wk = lead(hrs.wrk.wk), # Records hours worked per week worked in the subsequent year

lead.wage = lead(wages), lead.age = lead(age), # Recording wage, age, and region in the subsequent year

lead.region = lead(region)) %>%

ungroup() %>%

mutate(agriculture = case\_when(# Coding as agriculture when:

year %in% c(1980, 1990, 1999) & complete.cases(ind.orig) & ind.orig == 11 ~ 1, # Early covariate year observed, 2-digit 1970 PSID ind code is agriculture

year %in% c(1980, 1990, 1999) & is.na(ind.orig) & lead.ind1990 %in% c(10, 11, 30, 31, 32, 230) ~ 1, # Early covariate year unobserved, early wage year 1990 industry is agriculture

year %in% c(2009, 2015) & complete.cases(ind1990) & ind1990 %in% c(10, 11, 30, 31, 32, 230) ~ 1, # Late covariate year observed, 1990 ind code is agriculture

year %in% c(2009, 2015) & is.na(ind1990) & lead.ind1990 %in% c(10, 11, 30, 31, 32, 230) ~ 1, # Late covariate year unobserved, late wage year 1990 ind code is agriculture

year %in% c(1980, 1990, 1999) & complete.cases(occ.orig) & occ.orig %in% c(71, 80) ~ 1, # Early covariate year observed, 2-digit 1970 PSID occ code is farm laboreres & foremen, farmers & farm managers

year %in% c(1980, 1990, 1999) & is.na(occ.orig) & lead.occ2010 > 5940 & lead.occ2010 <= 6130 ~ 1, # Early covariate year unobserved, early wage year 2010 occ code is farm, forestry, fisheries

year %in% c(2009, 2015, 2107) & complete.cases(occ2010) & occ2010 > 5940 & occ2010 <= 6130 ~ 1, # Late covariate year observed, 2010 occ code is farm, forestry, fisheries

year %in% c(2009, 2015, 2017) & is.na(occ2010) & lead.occ2010 > 5940 & lead.occ2010 <= 6130 ~ 1, # Late covariate year unobserved, late wage year 2010 occ code is farm, forestry, fisheries

TRUE ~ 0),

agriculture = ifelse(year %in% c(1980, 1990, 1999, 2009, 2015, 2017), agriculture, NA),

# Following similar procedure as for agriculture for military

military = case\_when(

year %in% c(1980, 1990, 1999) & complete.cases(ind.orig) & ind.orig == 91 ~ 1,

year %in% c(1980, 1990, 1999) & is.na(ind.orig) & lead.ind1990 %in% c(940, 941, 942, 950, 951, 952, 960) ~ 1,

year %in% c(2009, 2015) & complete.cases(ind1990) & ind1990 %in% c(940, 941, 942, 950, 951, 952, 960) ~ 1,

year %in% c(2009, 2015) & is.na(ind1990) & lead.ind1990 %in% c(940, 941, 942, 950, 951, 952, 960) ~ 1,

year %in% c(1980, 1990, 1999) & complete.cases(occ.orig) & occ.orig == 55 ~ 1,

year %in% c(1980, 1990, 1999) & is.na(occ.orig) & lead.occ2010 > 9750 & lead.occ2010 <= 9830 ~ 1,

year %in% c(2009, 2015, 2017) & complete.cases(occ2010) & occ2010 > 9750 & occ2010 <= 9830 ~ 1,

year %in% c(2009, 2015, 2017) & is.na(occ2010) & lead.occ2010 > 9750 & lead.occ2010 <= 9830 ~ 1,

TRUE ~ 0),

military = ifelse(year %in% c(1980, 1990, 1999, 2009, 2015, 2017), military, NA),

# Creating variables for 2 digit ind & occ. These values are based on PSID 1970 codes for 1980,

# when not all individuals were recoded into the 1970 census schemes

# See technical appendix for detailed coding schemes with labels

ind.2d = case\_when(

ind1990 %in% c(10, 11, 30, 31, 32, 230) & year != 1980 |

ind.orig == 11 & year == 1980 ~ "Agriculture",

ind1990 %in% c(40, 41, 42, 50, 60) & year != 1980 |

ind.orig %in% c(21, 51) & year == 1980 ~ "Mining.Construction",

ind1990 %in% c(450, 451, 452, 470, 471, 472) & year != 1980 |

ind.orig == 57 & year == 1980 ~ "Utilities",

ind1990 %in% c(100, 101, 102, 110, 111, 112, 120, 121, 122, 130, 132, 140, 141,

142, 150, 151, 152, 160, 161, 162, 171, 172, 180, 181, 182, 190, 191,

192, 200, 201, 210, 211, 212, 220, 221, 222) & year != 1980 |

ind.orig %in% c(40:46, 85) & year == 1980 ~ "NonDurable.Manufacturing",

ind1990 %in% c(230, 231, 232, 241, 242, 250, 251, 252, 261, 262, 270, 271, 272, 280,

281, 282, 290, 291, 292, 300, 301, 310, 311, 312, 320, 321, 322, 331,

332, 340, 341, 342, 350, 351, 352, 360, 361, 362, 370, 371, 372, 380,

381, 390, 391, 392) & year != 1980 |

ind.orig %in% c(30:34, 49) & year == 1980 ~ "Durable.Manufacturing",

ind1990 %in% c(500, 501, 502, 510, 511, 512, 521, 530, 531, 532, 540, 541, 542, 550,

551, 552, 560, 561, 562, 571) & year != 1980 |

ind.orig %in% c(62, 69) & year == 1980 ~ "Wholesale.Trade",

ind1990 %in% c(580, 581, 582, 590, 591, 592, 600, 601, 602, 610, 611, 612, 620, 621,

622, 623, 630, 631, 632, 633, 640, 641, 642, 650, 651, 652, 660, 661,

662, 663, 670, 671, 672, 681, 682, 691) & year != 1980 |

ind.orig == 61 & year == 1980 ~ "Retail.Trade",

ind1990 %in% c(400, 401, 402, 410, 411, 412, 420, 421, 422, 432) & year != 1980 |

ind.orig == 55 & year == 1980 ~ "Transportation",

# Including movie theaters, libraries, data processing as communications (following Blau & Kahn),

ind1990 %in% c(852, 800, 732, 440, 441, 442) & year != 1980 |

ind.orig == 56 & year == 1980 ~ "Communications",

ind1990 %in% c(700, 701, 702, 710, 711, 712) & year != 1980 |

ind.orig == 71 & year == 1980 ~ "Finance",

ind1990 %in% c(842, 850, 851, 860) & year != 1980 |

ind.orig == 87 & year == 1980 ~ "Education",

ind1990 %in% c(870, 812, 820, 821, 822, 830, 831, 832, 840) & year != 1980 |

ind.orig == 86 & year == 1980 ~ "Medical",

ind1990 %in% c(841, 721, 722, 731, 740, 741, 861, 882, 890, 891, 892, 893, 20, 12) & year != 1980 |

ind.orig %in% c(82, 88) & year == 1980 ~ "Professional",

ind1990 %in% c(900, 901, 910, 921, 922, 930, 931, 932) & year != 1980 |

ind.orig == 92 & year == 1980 ~ "Public.Administration",

ind1990 %in% c(940, 941, 942, 950, 951, 952, 960) & year != 1980 |

ind.orig == 91 & year == 1980 ~ "Active.Duty.Military",

ind1990 %in% c(742, 750, 751, 752, 760, 761, 762, 770, 771, 772, 780, 781, 782, 790, 791, 801, 802, 810,

862, 863, 871, 872, 873, 880, 881) & year != 1980 |

ind.orig %in% c(81, 83, 84) & year == 1980 ~ "Social.Work.Arts.Recreation.Other.Services"))

# Creating data on ever reported race, using heads & wives' individual race and ethnicity

# reports starting in 1985. Prior to 1985, race data was only collected on household heads: race

# was assumed to be the same for spouses/wives, as well as for splitoff families if race was carried

# over from a prior year and not re-asked in the interview year. The following section gathers all

# respondents who are ever heads/wives and report race starting in 1985

psid\_race <- read.dta("Raw Data/psid/psid\_wrk.dta") %>%

transmute(

family\_id = ER30001, # 1968 interview number

person\_number = ER30002, # 1968 person number

indiv.id = paste(family\_id, person\_number, sep ="\_"), # unique individual identifier

rel.head\_1985 = ER30465, rel.head\_1986 = ER30500, rel.head\_1987 = ER30537, # Relationship to head

rel.head\_1988 = ER30572, rel.head\_1989 = ER30608, rel.head\_1990 = ER30644,

rel.head\_1991 = ER30691, rel.head\_1992 = ER30735, rel.head\_1993 = ER30808,

rel.head\_1994 = ER33103, rel.head\_1995 = ER33203, rel.head\_1996 = ER33303,

rel.head\_1997 = ER33403, rel.head\_1999 = ER33503, rel.head\_2001 = ER33603,

rel.head\_2003 = ER33703, rel.head\_2005 = ER33803, rel.head\_2007 = ER33903,

rel.head\_2009 = ER34003, rel.head\_2011 = ER34103, rel.head\_2013 = ER34203,

rel.head\_2015 = ER34303, rel.head\_2017 = ER34503, rel.head\_2019 =ER34703,

# Following Blau & Kahn's coding procedure, we use individual's reported "Spanish" ethnicity/descent

# to code individuals as Hispanic if they report having Hispanic descent; we code individuals as

# Black if they report being black in any race question asked that year, and as other if they

# report an "other" race in any race questions asked that year. We code the remaining individuals

# as white if they report being White and do not report "having Hispanic ancestry, do not report

# being Black in any race question, and do not report an "other" race in any race question.

# From 1985 to 1993, individuals can report up to two races. In 1994 to 1996, individuals can

# report up to three races. Starting in 1997 and up to 2003, individuals can report up to four

# races, but "Hispanic" is not asked separately and is an option in the race questions. Starting

# in 2005 up to 2017, the PSID asks about Hispanic ancestry separately and individuals can report

# up to four races

race.hd\_1985 = case\_when(V11937 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V11938 == 2 | V11939 == 2 ~ "Black",

V11938 %in% c(3,4,7) | V11939 %in% c(3,4,7) ~ "Other",

V11938 == 1 | V11939 == 1 ~ "White"),

race.wf\_1985 = case\_when(V12292 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V12293 == 2 | V12294 == 2 ~ "Black",

V12293 %in% c(3,4,7) | V12294 %in% c(3,4,7) ~ "Other",

V12293 == 1 | V12294 == 1 ~ "White"),

race.hd\_1986 = case\_when(V13564 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V13565 == 2 | V13566 == 2 ~ "Black",

V13565 %in% c(3,4,7) | V13566 %in% c(3,4,7) ~ "Other",

V13565 == 1 | V13566 == 1 ~ "White"),

race.wf\_1986 = case\_when(V13499 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V13500 == 2 | V13501 == 2 ~ "Black",

V13500 %in% c(3,4,7) | V13501 %in% c(3,4,7) ~ "Other",

V13500 == 1 | V13501 == 1 ~ "White"),

race.hd\_1987 = case\_when(V14611 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V14612 == 2 | V14613 == 2 ~ "Black",

V14612 %in% c(3,4,7) | V14613 %in% c(3,4,7) ~ "Other",

V14612 == 1 | V14613 == 1 ~ "White"),

race.wf\_1987 = case\_when(V14546 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V14547 == 2 | V14548 == 2 ~ "Black",

V14547 %in% c(3,4,7) | V14548 %in% c(3,4,7) ~ "Other",

V14547 == 1 | V14548 == 1 ~ "White"),

race.hd\_1988 = case\_when(V16085 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V16086 == 2 | V16087 == 2 ~ "Black",

V16086 %in% c(3,4,7) | V16087 %in% c(3,4,7) ~ "Other",

V16086 == 1 | V16087 == 1 ~ "White"),

race.wf\_1988 = case\_when(V16020 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V16021 == 2 | V16022 == 2 ~ "Black",

V16021 %in% c(3,4,7) | V16022 %in% c(3,4,7) ~ "Other",

V16021 == 1 | V16022 == 1 ~ "White"),

race.hd\_1989 = case\_when(V17482 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V17483 == 2 | V17484 == 2 ~ "Black",

V17483 %in% c(3,4,7) | V17484 %in% c(3,4,7) ~ "Other",

V17483 == 1 | V17484 == 1 ~ "White"),

race.wf\_1989 = case\_when(V17417 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V17418 == 2 | V17419 == 2 ~ "Black",

V17418 %in% c(3,4,7) | V17419 %in% c(3,4,7) ~ "Other",

V17418 == 1 | V17419 == 1 ~ "White"),

race.hd\_1990 = case\_when(V18813 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V18814 == 2 | V18815 == 2 ~ "Black",

V18814 %in% c(3,4,7) | V18815 %in% c(3,4,7) ~ "Other",

V18814 == 1 | V18815 == 1 ~ "White"),

race.wf\_1990 = case\_when(V18748 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V18749 == 2 | V18750 == 2 ~ "Black",

V18749 %in% c(3,4,7) | V18750 %in% c(3,4,7) ~ "Other",

V18749 == 1 | V18750 == 1 ~ "White"),

race.hd\_1991 = case\_when(V20113 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V20114 == 2 | V20115 == 2 ~ "Black",

V20114 %in% c(3,4,7) | V20115 %in% c(3,4,7) ~ "Other",

V20114 == 1 | V20115 == 1 ~ "White"),

race.wf\_1991 = case\_when(V20048 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V20049 == 2 | V20050 == 2 ~ "Black",

V20049 %in% c(3,4,7) | V20050 %in% c(3,4,7) ~ "Other",

V20049 == 1 | V20050 == 1 ~ "White"),

race.hd\_1992 = case\_when(V21419 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V21420 == 2 | V21421 == 2 ~ "Black",

V21420 %in% c(3,4,7) | V21421 %in% c(3,4,7) ~ "Other",

V21420 == 1 | V21421 == 1 ~ "White"),

race.wf\_1992 = case\_when(V21354 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V21355 == 2 | V21356 == 2 ~ "Black",

V21355 %in% c(3,4,7) | V21356 %in% c(3,4,7) ~ "Other",

V21355 == 1 | V21356 == 1 ~ "White"),

race.hd\_1993 = case\_when(V23275 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V23276 == 2 | V23277 == 2 ~ "Black",

V23276 %in% c(3,4,7) | V23277 %in% c(3,4,7) ~ "Other",

V23276 == 1 | V23277 == 1 ~ "White"),

race.wf\_1993 = case\_when(V23211 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

V23212 == 2 | V23213 == 2 ~ "Black",

V23212 %in% c(3,4,7) | V23213 %in% c(3,4,7) ~ "Other",

V23212 == 1 | V23213 == 1 ~ "White"),

race.hd\_1994 = case\_when(ER3941 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER3944 == 2 | ER3945 == 2 | ER3946 == 2 ~ "Black",

ER3944 %in% c(3,4,7) | ER3945 %in% c(3,4,7) | ER3946 %in% c(3,4,7) ~ "Other",

ER3944 == 1 | ER3945 == 1 | ER3946 == 1 ~ "White"),

race.wf\_1994 = case\_when(ER3880 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER3883 == 2 | ER3884 == 2 | ER3885 == 2 ~ "Black",

ER3883 %in% c(3,4,7) | ER3884 %in% c(3,4,7) | ER3885 %in% c(3,4,7) ~ "Other",

ER3883 == 1 | ER3884 == 1 | ER3885 == 1 ~ "White"),

race.hd\_1995 = case\_when(ER6811 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER6814 == 2 | ER6815 == 2 | ER6816 == 2 ~ "Black",

ER6814 %in% c(3,4,7) | ER6815 %in% c(3,4,7) | ER6816 %in% c(3,4,7) ~ "Other",

ER6814 == 1 | ER6815 == 1 | ER6816 == 1 ~ "White"),

race.wf\_1995 = case\_when(ER6750 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER6753 == 2 | ER6754 == 2 | ER6755 == 2 ~ "Black",

ER6753 %in% c(3,4,7) | ER6754 %in% c(3,4,7) | ER6755 %in% c(3,4,7) ~ "Other",

ER6753 == 1 | ER6754 == 1 | ER6755 == 1 ~ "White"),

race.hd\_1996 = case\_when(ER9057 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER9060 == 2 | ER9061 == 2 | ER9062 == 2 ~ "Black",

ER9060 %in% c(3,4,7) | ER9061 %in% c(3,4,7) | ER9062 %in% c(3,4,7) ~ "Other",

ER9060 == 1 | ER9061 == 1 | ER9062 == 1 ~ "White"),

race.wf\_1996 = case\_when(ER8996 %in% c(1, 2, 3, 4, 5, 6, 7) ~ "Hispanic",

ER8999 == 2 | ER9000 == 2 | ER9001 == 2 ~ "Black",

ER8999 %in% c(3,4,7) | ER9000 %in% c(3,4,7) | ER9001 %in% c(3,4,7) ~ "Other",

ER8999 == 1 | ER9000 == 1 | ER9001 == 1 ~ "White"),

race.hd\_1997 = case\_when(ER11848 == 5 | ER11849 == 5 | ER11850 == 5 | ER11851 == 5 ~ "Hispanic",

ER11848 == 2 | ER11849 == 2 | ER11850 == 2 | ER11851 == 2 ~ "Black",

ER11848 %in% c(3,4,6,7) | ER11849 %in% c(3,4,6,7) | ER11850 %in% c(3,4,6,7) | ER11851 %in% c(3,4,6,7) ~ "Other",

ER11848 == 1 | ER11849 == 1 | ER11850 == 1 | ER11851 == 1 ~ "White"),

race.wf\_1997 = case\_when(ER11760 == 5 | ER11761 == 5 | ER11762 == 5 | ER11763 == 5 ~ "Hispanic",

ER11760 == 2 | ER11761 == 2 | ER11762 == 2 | ER11763 == 2 ~ "Black",

ER11760 %in% c(3,4,6,7) | ER11761 %in% c(3,4,6,7) | ER11762 %in% c(3,4,6,7) | ER11763 %in% c(3,4,6,7) ~ "Other",

ER11760 == 1 | ER11761 == 1 | ER11762 == 1 | ER11763 == 1 ~ "White"),

race.hd\_1999 = case\_when(ER15928 == 5 | ER15929 == 5 | ER15930 == 5 | ER15931 == 5 ~ "Hispanic",

ER15928 == 2 | ER15929 == 2 | ER15930 == 2 | ER15931 == 2 ~ "Black",

ER15928 %in% c(3,4,6,7) | ER15929 %in% c(3,4,6,7) | ER15930 %in% c(3,4,6,7) | ER15931 %in% c(3,4,6,7) ~ "Other",

ER15928 == 1 | ER15929 == 1 | ER15930 == 1 | ER15931 == 1 ~ "White"),

race.wf\_1999 = case\_when(ER15836 == 5 | ER15837 == 5 | ER15838 == 5 | ER15839 == 5 ~ "Hispanic",

ER15836 == 2 | ER15837 == 2 | ER15838 == 2 | ER15839 == 2 ~ "Black",

ER15836 %in% c(3,4,6,7) | ER15837 %in% c(3,4,6,7) | ER15838 %in% c(3,4,6,7) | ER15839 %in% c(3,4,6,7) ~ "Other",

ER15836 == 1 | ER15837 == 1 | ER15838 == 1 | ER15839 == 1 ~ "White"),

race.hd\_2001 = case\_when(ER19989 == 5 | ER19990 == 5 | ER19991 == 5 | ER19992 == 5 ~ "Hispanic",

ER19989 == 2 | ER19990 == 2 | ER19991 == 2 | ER19992 == 2 ~ "Black",

ER19989 %in% c(3,4,6,7) | ER19990 %in% c(3,4,6,7) | ER19991 %in% c(3,4,6,7) | ER19992 %in% c(3,4,6,7) ~ "Other",

ER19989 == 1 | ER19990 == 1 | ER19991 == 1 | ER19992 == 1 ~ "White"),

race.wf\_2001 = case\_when(ER19897 == 5 | ER19898 == 5 | ER19899 == 5 | ER19900 == 5 ~ "Hispanic",

ER19897 == 2 | ER19898 == 2 | ER19899 == 2 | ER19900 == 2 ~ "Black",

ER19897 %in% c(3,4,6,7) | ER19898 %in% c(3,4,6,7) | ER19899 %in% c(3,4,6,7) | ER19900 %in% c(3,4,6,7) ~ "Other",

ER19897 == 1 | ER19898 == 1 | ER19899 == 1 | ER19900 == 1 ~ "White"),

race.hd\_2003 = case\_when(ER23426 == 5 | ER23427 == 5 | ER23428 == 5 | ER23429 == 5 ~ "Hispanic",

ER23426 == 2 | ER23427 == 2 | ER23428 == 2 | ER23429 == 2 ~ "Black",

ER23426 %in% c(3,4,6,7) | ER23427 %in% c(3,4,6,7) | ER23428 %in% c(3,4,6,7) | ER23429 %in% c(3,4,6,7) ~ "Other",

ER23426 == 1 | ER23427 == 1 | ER23428 == 1 | ER23429 == 1 ~ "White"),

race.wf\_2003 = case\_when(ER23334 == 5 | ER23335 == 5 | ER23336 == 5 | ER23337 == 5 ~ "Hispanic",

ER23334 == 2 | ER23335 == 2 | ER23336 == 2 | ER23337 == 2 ~ "Black",

ER23334 %in% c(3,4,6,7) | ER23335 %in% c(3,4,6,7) | ER23336 %in% c(3,4,6,7) | ER23337 %in% c(3,4,6,7) ~ "Other",

ER23334 == 1 | ER23335 == 1 | ER23336 == 1 | ER23337 == 1 ~ "White"),

race.hd\_2005 = case\_when(ER27392 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER27393 == 2 | ER27394 == 2 | ER27395 == 2 | ER27396 == 2 ~ "Black",

ER27393 %in% c(3,4,5,7) | ER27394 %in% c(3,4,5,7) | ER27395 %in% c(3,4,5,7) | ER27396 %in% c(3,4,5,7) ~ "Other",

ER27393 == 1 | ER27394 == 1 | ER27395 == 1 | ER27396 == 1 ~ "White"),

race.wf\_2005 = case\_when(ER27296 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER27297 == 2 | ER27298 == 2 | ER27299 == 2 | ER27300 == 2 ~ "Black",

ER27297 %in% c(3,4,5,7) | ER27298 %in% c(3,4,5,7) | ER27299 %in% c(3,4,5,7) | ER27300 %in% c(3,4,5,7) ~ "Other",

ER27297 == 1 | ER27298 == 1 | ER27299 == 1 | ER27300 == 1 ~ "White"),

race.hd\_2007 = case\_when(ER40564 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER40565 == 2 | ER40566 == 2 | ER40567 == 2 | ER40568 == 2 ~ "Black",

ER40565 %in% c(3,4,5,7) | ER40566 %in% c(3,4,5,7) | ER40567 %in% c(3,4,5,7) | ER40568 %in% c(3,4,5,7) ~ "Other",

ER40565 == 1 | ER40566 == 1 | ER40567 == 1 | ER40568 == 1 ~ "White"),

race.wf\_2007 = case\_when(ER40471 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER40472 == 2 | ER40473 == 2 | ER40474 == 2 | ER40475 == 2 ~ "Black",

ER40472 %in% c(3,4,5,7) | ER40473 %in% c(3,4,5,7) | ER40474 %in% c(3,4,5,7) | ER40475 %in% c(3,4,5,7) ~ "Other",

ER40472 == 1 | ER40473 == 1 | ER40474 == 1 | ER40475 == 1 ~ "White"),

race.hd\_2009 = case\_when(ER46542 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER46543 == 2 | ER46544 == 2 | ER46545 == 2 | ER46546 == 2 ~ "Black",

ER46543 %in% c(3,4,5,7) | ER46544 %in% c(3,4,5,7) | ER46545 %in% c(3,4,5,7) | ER46546 %in% c(3,4,5,7) ~ "Other",

ER46543 == 1 | ER46544 == 1 | ER46545 == 1 | ER46546 == 1 ~ "White"),

race.wf\_2009 = case\_when(ER46448 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER46449 == 2 | ER46450 == 2 | ER46451 == 2 | ER46452 == 2 ~ "Black",

ER46449 %in% c(3,4,5,7) | ER46450 %in% c(3,4,5,7) | ER46451 %in% c(3,4,5,7) | ER46451 %in% c(3,4,5,7) ~ "Other",

ER46449 == 1 | ER46450 == 1 | ER46451 == 1 | ER46452 == 1 ~ "White"),

race.hd\_2011 = case\_when(ER51903 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER51904 == 2 | ER51905 == 2 | ER51906 == 2 | ER51907 == 2 ~ "Black",

ER51904 %in% c(3,4,5,7) | ER51905 %in% c(3,4,5,7) | ER51906 %in% c(3,4,5,7) | ER51907 %in% c(3,4,5,7) ~ "Other",

ER51904 == 1 | ER51905 == 1 | ER51906 == 1 | ER51907 == 1 ~ "White"),

race.wf\_2011 = case\_when(ER51809 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER51810 == 2 | ER51811 == 2 | ER51812 == 2 | ER51813 == 2 ~ "Black",

ER51810 %in% c(3,4,5,7) | ER51811 %in% c(3,4,5,7) | ER51812 %in% c(3,4,5,7) | ER51813 %in% c(3,4,5,7) ~ "Other",

ER51810 == 1 | ER51811 == 1 | ER51812 == 1 | ER51813 == 1 ~ "White"),

race.hd\_2013 = case\_when(ER57658 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER57659 == 2 | ER57660 == 2 | ER57661 == 2 | ER57662 == 2 ~ "Black",

ER57659 %in% c(3,4,5,7) | ER57660 %in% c(3,4,5,7) | ER57661 %in% c(3,4,5,7) | ER57662 %in% c(3,4,5,7) ~ "Other",

ER57659 == 1 | ER57660 == 1 | ER57661 == 1 | ER57662 == 1 ~ "White"),

race.wf\_2013 = case\_when(ER57548 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER57549 == 2 | ER57550 == 2 | ER57551 == 2 | ER57552 == 2 ~ "Black",

ER57549 %in% c(3,4,5,7) | ER57550 %in% c(3,4,5,7) | ER57551 %in% c(3,4,5,7) | ER57552 %in% c(3,4,5,7) ~ "Other",

ER57549 == 1 | ER57550 == 1 | ER57551 == 1 | ER57552 == 1 ~ "White"),

race.hd\_2015 = case\_when(ER64809 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER64810 == 2 | ER64811 == 2 | ER64812 == 2 | ER64813 == 2 ~ "Black",

ER64810 %in% c(3,4,5,7) | ER64811 %in% c(3,4,5,7) | ER64812 %in% c(3,4,5,7) | ER64813 %in% c(3,4,5,7) ~ "Other",

ER64810 == 1 | ER64811 == 1 | ER64812 == 1 | ER64813 == 1 ~ "White"),

race.wf\_2015 = case\_when(ER64670 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER64671 == 2 | ER64672 == 2 | ER64673 == 2 | ER64674 == 2 ~ "Black",

ER64671 %in% c(3,4,5,7) | ER64672 %in% c(3,4,5,7) | ER64673 %in% c(3,4,5,7) | ER64674 %in% c(3,4,5,7) ~ "Other",

ER64671 == 1 | ER64672 == 1 | ER64673 == 1 | ER64674 == 1 ~ "White"),

race.hd\_2017 = case\_when(ER70881 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER70882 == 2 | ER70883 == 2 | ER70884 == 2 | ER70885 == 2 ~ "Black",

ER70882 %in% c(3,4,5,7) | ER70883 %in% c(3,4,5,7) | ER70884 %in% c(3,4,5,7) | ER70885 %in% c(3,4,5,7) ~ "Other",

ER70882 == 1 | ER70883 == 1 | ER70884 == 1 | ER70885 == 1 ~ "White"),

race.wf\_2017 = case\_when(ER70743 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER70744 == 2 | ER70745 == 2 | ER70746 == 2 | ER70747 == 2 ~ "Black",

ER70744 %in% c(3,4,5,7) | ER70745 %in% c(3,4,5,7) | ER70746 %in% c(3,4,5,7) | ER70747 %in% c(3,4,5,7) ~ "Other",

ER70744 == 1 | ER70745 == 1 | ER70746 == 1 | ER70747 == 1 ~ "White"),

race.hd\_2019 = case\_when(ER76896 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER76897 == 2 | ER76898 == 2 | ER76899 == 2 | ER76900 == 2 ~ "Black",

ER76897 %in% c(3,4,5,7) | ER76898 %in% c(3,4,5,7) | ER76899 %in% c(3,4,5,7) | ER76900 %in% c(3,4,5,7) ~ "Other",

ER76897 == 1 | ER76898 == 1 | ER76899 == 1 | ER76900 == 1 ~ "White"),

race.wf\_2019 = case\_when(ER76751 %in% c(1, 2, 3, 4, 5, 7) ~ "Hispanic",

ER76752 == 2 | ER76753 == 2 | ER76754 == 2 | ER76755 == 2 ~ "Black",

ER76752 %in% c(3,4,5,7) | ER76753 %in% c(3,4,5,7) | ER76754 %in% c(3,4,5,7) | ER76755 %in% c(3,4,5,7) ~ "Other",

ER76752 == 1 | ER76753 == 1 | ER76754 == 1 | ER76755 == 1 ~ "White")

) %>%

gather(key, value, -c(family\_id, person\_number, indiv.id)) %>%

separate(key, into = c("key", "year"), sep = "\_") %>%

# Grouping by individual id

group\_by(indiv.id) %>%

# Turning data back to wide format, each record is a person-year

spread(key, value, convert = T) %>%

mutate(race = case\_when(rel.head == 10 ~ race.hd, # Coding race as race of head if R is head,

rel.head %in% c(20, 22) ~ race.wf)) %>% # race of wive if R is wife

dplyr::select(indiv.id, year, race) %>%

# This gives us a wide dataset where each row is an individual & each column is the coded race for that year

spread(key = year, value = race)

# Creating a new dataframe that codes a respondent's race based on all of the answers hey've ever reported

psid\_race\_ever <- unite(psid\_race, newCol, -indiv.id) %>% # Creates a variable that strings together each year's race report

# Codes respondent's race as Hispanic if they ever report Hispanic ancestry, Black if they

# are ever coded as Black, Other if they are ever coded as Other, and White if they are never coded as

# any of the above but are ever coded as White. The remainder are set to missing implicitly by case\_when

mutate(race.ever = case\_when(grepl("Hispanic", newCol) ~ "Hispanic",

grepl("Black", newCol) ~ "Black",

grepl("Other", newCol) ~ "Other",

grepl("White", newCol) ~ "White")) %>%

dplyr::select(-newCol)

# Joining ever reported race to the observed data

psid\_obs <- psid\_clean %>%

left\_join(., psid\_race\_ever, by = "indiv.id") %>%

# Coding a respondent's race as the race they ever reported if available (~90%): if not,

# R is coded as race of the head (for ex. if respondent attrits prior to 1985 but is in

# the data in 1981)

mutate(race = ifelse(is.na(race.ever), racehd, race.ever))

# Generating the work experience variables using yearly hours worked & reported experience variables

psid\_exp <- read.dta("Raw Data/psid/psid\_wrk.dta") %>%

# Selecting relevant variables & renaming by year

transmute(

intnum68 = ER30001, pernum68 = ER30002,female = ifelse(ER32000 == 2, 1, 0),

int.num\_1976 = ER30188, seq.num\_1976 = ER30189, rel.head\_1976 = ER30190, age\_1976 = ER30191,

yearswrk\_hd\_1976 = V4630, yearswrk.ft\_hd\_1976 = V4631, yearswrk\_wf\_1976 = V4989, yearswrk.ft\_wf\_1976 = V4990,

int.num\_1977 = ER30217, seq.num\_1977 = ER30218, rel.head\_1977 = ER30219, age\_1977 = ER30220,

ann.wrk.hrs\_hd\_1977 = V5232, ann.wrk.hrs\_wf\_1977 = V5244,

yearswrk\_hd\_1977 = V5604, yearswrk.ft\_hd\_1977 = V5605, yearswrk\_wf\_1977 = V5574, yearswrk.ft\_wf\_1977 = V5575,

int.num\_1978 = ER30246, seq.num\_1978 = ER30247, rel.head\_1978 = ER30248, age\_1978 = ER30249,

ann.wrk.hrs\_hd\_1978 = V5731, ann.wrk.hrs\_wf\_1978 = V5743,

yearswrk\_hd\_1978 = V6153, yearswrk.ft\_hd\_1978 = V6154, yearswrk\_wf\_1978 = V6123, yearswrk.ft\_wf\_1978 = V6124,

int.num\_1979 = ER30283, seq.num\_1979 = ER30284, rel.head\_1979 = ER30285, age\_1979 = ER30286,

age\_1979 = ER30286, ann.wrk.hrs\_hd\_1979 = V6336, ann.wrk.hrs\_wf\_1979 = V6348,

yearswrk\_hd\_1979 = V6750, yearswrk.ft\_hd\_1979 = V6751, yearswrk\_wf\_1979 = V6720, yearswrk.ft\_wf\_1979 = V6721,

int.num\_1980 = ER30313, seq.num\_1980 = ER30314, rel.head\_1980 = ER30315, age\_1980 = ER30316,

age\_1980 = ER30316, ann.wrk.hrs\_hd\_1980 = V6934, ann.wrk.hrs\_wf\_1980 = V6946,

yearswrk\_hd\_1980 = V7383, yearswrk.ft\_hd\_1980 = V7384, yearswrk\_wf\_1980 = V7353, yearswrk.ft\_wf\_1980 = V7354,

int.num\_1981 = ER30343, seq.num\_1981 = ER30344, rel.head\_1981 = ER30345, age\_1981 = ER30346,

age\_1981 = ER30346, ann.wrk.hrs\_hd\_1981 = V7530, ann.wrk.hrs\_wf\_1981 = V7540,

yearswrk\_hd\_1981 = V7383, yearswrk.ft\_hd\_1981 = V7384, yearswrk\_wf\_1981 = V7353, yearswrk.ft\_wf\_1981 = V7354,

int.num\_1982 = ER30373, seq.num\_1982 = ER30374, rel.head\_1982 = ER30375, age\_1982 = ER30376,

ann.wrk.hrs\_hd\_1982 = V8228, ann.wrk.hrs\_wf\_1982 = V8238,

yearswrk\_hd\_1982 = V8659, yearswrk.ft\_hd\_1982 = V8660, yearswrk\_wf\_1982 = V8629, yearswrk.ft\_wf\_1982 = V8630,

int.num\_1983 = ER30399, seq.num\_1983 = ER30400, rel.head\_1983 = ER30401, age\_1983 = ER30402,

ann.wrk.hrs\_hd\_1983 = V8830, ann.wrk.hrs\_wf\_1983 = V8840,

yearswrk\_hd\_1983 = V9345, yearswrk.ft\_hd\_1983 = V9346, yearswrk\_wf\_1983 = V9315, yearswrk.ft\_wf\_1983 = V9316,

int.num\_1984 = ER30429, seq.num\_1984 = ER30430, rel.head\_1984 = ER30431, age\_1984 = ER30432,

ann.wrk.hrs\_hd\_1984 = V10037, ann.wrk.hrs\_wf\_1984 = V10131,

yearswrk\_hd\_1984 = V10992, yearswrk.ft\_hd\_1984 = V10993, yearswrk\_wf\_1984 = V10962, yearswrk.ft\_wf\_1984 = V10963,

int.num\_1985 = ER30463, seq.num\_1985 = ER30464, rel.head\_1985 = ER30465, age\_1985 = ER30466,

ann.wrk.hrs\_hd\_1985 = V11146, ann.wrk.hrs\_wf\_1985 = V11258,

yearswrk\_hd\_1985 = V11739, yearswrk.ft\_hd\_1985 = V11740, yearswrk\_wf\_1985 = V12102, yearswrk.ft\_wf\_1985 = V12103,

int.num\_1986 = ER30498, seq.num\_1986 = ER30499, rel.head\_1986 = ER30500, age\_1986 = ER30501,

ann.wrk.hrs\_hd\_1986 = V12545, ann.wrk.hrs\_wf\_1986 = V12657,

yearswrk\_hd\_1986 = V13605, yearswrk.ft\_hd\_1986 = V13606, yearswrk\_wf\_1986 = V13531, yearswrk.ft\_wf\_1986 = V13532,

int.num\_1987 = ER30535, seq.num\_1987 = ER30536, rel.head\_1987 = ER30537, age\_1987 = ER30538,

ann.wrk.hrs\_hd\_1987 = V13745, ann.wrk.hrs\_wf\_1987 = V13809,

yearswrk\_hd\_1987 = V14652, yearswrk.ft\_hd\_1987 = V14653, yearswrk\_wf\_1987 = V14578, yearswrk.ft\_wf\_1987 = V14579,

int.num\_1988 = ER30570, seq.num\_1988 = ER30571, rel.head\_1988 = ER30572, age\_1988 = ER30573,

ann.wrk.hrs\_hd\_1988 = V14835, ann.wrk.hrs\_wf\_1988 = V14865,

yearswrk\_hd\_1988 = V16126, yearswrk.ft\_hd\_1988 = V16127, yearswrk\_wf\_1988 = V16052, yearswrk.ft\_wf\_1988 = V16053,

int.num\_1989 = ER30606, seq.num\_1989 = ER30607, rel.head\_1989 = ER30608, age\_1989 = ER30609,

ann.wrk.hrs\_hd\_1989 = V16335, ann.wrk.hrs\_wf\_1989 = V16365,

yearswrk\_hd\_1989 = V17523, yearswrk.ft\_hd\_1989 = V17524, yearswrk\_wf\_1989 = V17449, yearswrk.ft\_wf\_1989 = V17450,

int.num\_1990 = ER30642, seq.num\_1990 = ER30643, rel.head\_1990 = ER30644, age\_1990 = ER30645,

ann.wrk.hrs\_hd\_1990 = V17744, ann.wrk.hrs\_wf\_1990 = V17774,

yearswrk\_hd\_1990 = V18854, yearswrk.ft\_hd\_1990 = V18855, yearswrk\_wf\_1990 = V18780, yearswrk.ft\_wf\_1990 = V18781,

int.num\_1991 = ER30689, seq.num\_1991 = ER30690, rel.head\_1991 = ER30691, age\_1991 = ER30692,

ann.wrk.hrs\_hd\_1991 = V19044, ann.wrk.hrs\_wf\_1991 = V19074,

yearswrk\_hd\_1991 = V20154, yearswrk.ft\_hd\_1991 = V20155, yearswrk\_wf\_1991 = V20080, yearswrk.ft\_wf\_1991 = V20081,

int.num\_1992 = ER30733, seq.num\_1992 = ER30734, rel.head\_1992 = ER30735, age\_1992 = ER30736,

ann.wrk.hrs\_hd\_1992 = V20344, ann.wrk.hrs\_wf\_1992 = V20374,

yearswrk\_hd\_1992 = V21460, yearswrk.ft\_hd\_1992 = V21461, yearswrk\_wf\_1992 = V21386, yearswrk.ft\_wf\_1992 = V21387,

int.num\_1993 = ER30806, seq.num\_1993 = ER30807, rel.head\_1993 = ER30808, age\_1993 = ER30809,

ann.wrk.hrs\_hd\_1993 = V21634, ann.wrk.hrs\_wf\_1993 = V21670,

yearswrk\_hd\_1993 = V23316, yearswrk.ft\_hd\_1993 = V23317, yearswrk\_wf\_1993 = V23243, yearswrk.ft\_wf\_1993 = V23244,

int.num\_1994 = ER33101, seq.num\_1994 = ER33102, rel.head\_1994 = ER33103, age\_1994 = ER33104,

ann.wrk.hrs\_hd\_1994 = ER4096, ann.wrk.hrs\_wf\_1994 = ER4107,

yearswrk\_hd\_1994 = ER3985, yearswrk.ft\_hd\_1994 = ER3986, yearswrk\_wf\_1994 = ER3915, yearswrk.ft\_wf\_1994 = ER3916,

int.num\_1995 = ER33201, seq.num\_1995 = ER33202, rel.head\_1995 = ER33203, age\_1995 = ER33204,

ann.wrk.hrs\_hd\_1995 = ER6936, ann.wrk.hrs\_wf\_1995 = ER6947,

yearswrk\_hd\_1995 = ER6855, yearswrk.ft\_hd\_1995 = ER6856, yearswrk\_wf\_1995 = ER6785, yearswrk.ft\_wf\_1995 = ER6786,

int.num\_1996 = ER33301, seq.num\_1996 = ER33302, rel.head\_1996 = ER33303, age\_1996 = ER33304,

ann.wrk.hrs\_hd\_1996 = ER9187, ann.wrk.hrs\_wf\_1996 = ER9198,

yearswrk\_hd\_1996 = ER9101, yearswrk.ft\_hd\_1996 = ER9102, yearswrk\_wf\_1996 = ER9031, yearswrk.ft\_wf\_1996 = ER9032,

int.num\_1997 = ER33401, seq.num\_1997 = ER33402, rel.head\_1997 = ER33403, age\_1997 = ER33404,

ann.wrk.hrs\_hd\_1997 = ER12174, ann.wrk.hrs\_wf\_1997 = ER12185,

yearswrk\_hd\_1997 = ER11897, yearswrk.ft\_hd\_1997 = ER11898, yearswrk\_wf\_1997 = ER11809, yearswrk.ft\_wf\_1997 = ER11810,

# Expand for missing- 1998

int.num\_1998 = ER33501, seq.num\_1998 = ER33502, rel.head\_1998 = ER33503, wkswrktot\_1998 = ER33536C, monthswrk\_1998 = ER33536P,

wkspermonth\_1998 = ER33536Q, yearswrk\_hd\_1998 = NA, yearswrk.ft\_hd\_1998 = NA, yearswrk\_wf\_1998 = NA, yearswrk.ft\_wf\_1998 = NA,

ann.wrk.hrs\_hd\_1998 = NA, ann.wrk.hrs\_wf\_1998 = NA,

int.num\_1999 = ER33501, seq.num\_1999 = ER33502, rel.head\_1999 = ER33503, age\_1999 = ER33504,

ann.wrk.hrs\_hd\_1999 = ER16471, ann.wrk.hrs\_wf\_1999 = ER16482,

yearswrk\_hd\_1999 = ER15979, yearswrk.ft\_hd\_1999 = ER15980, yearswrk\_wf\_1999 = ER15886, yearswrk.ft\_wf\_1999 = ER15887,

# Expand for missing- 2000

int.num\_2000 = ER33601, seq.num\_2000 = ER33602, rel.head\_2000 = ER33603, wkswrktot\_2000 = ER33627C, monthswrk\_2000 = ER33627P,

wkspermonth\_2000 = ER33627Q, yearswrk\_hd\_2000 = NA, yearswrk.ft\_hd\_2000 = NA, yearswrk\_wf\_2000 = NA, yearswrk.ft\_wf\_2000 = NA,

ann.wrk.hrs\_hd\_2000 = NA, ann.wrk.hrs\_wf\_2000 = NA,

int.num\_2001 = ER33601, seq.num\_2001 = ER33602, rel.head\_2001 = ER33603, age\_2001 = ER33604,

ann.wrk.hrs\_hd\_2001 = ER20399, ann.wrk.hrs\_wf\_2001 = ER20410,

yearswrk\_hd\_2001 = ER20040, yearswrk.ft\_hd\_2001 = ER20041, yearswrk\_wf\_2001 = ER19947, yearswrk.ft\_wf\_2001 = ER19948,

int.num\_2002 = ER33701, seq.num\_2002 = ER33702, rel.head\_2002 = ER33703, yearswrk\_hd\_2002 = NA, yearswrk.ft\_hd\_2002 = NA,

yearswrk\_wf\_2002 = NA, yearswrk.ft\_wf\_2002 = NA, ann.wrk.hrs\_hd\_2002 = NA, ann.wrk.hrs\_wf\_2002 = NA,

employed\_hd\_2002 = ER23702D2, employed\_wf\_2002 = ER23702J5, wkswrk\_hd\_2002 = ER23702D3, wkswrk\_wf\_2002 = ER23702J6,

hrs.per.wk\_hd\_2002 = ER23702E8, hrs.per.wk\_wf\_2002 = ER23702L2,

int.num\_2003 = ER33701, seq.num\_2003 = ER33702, rel.head\_2003 = ER33703, age\_2003 = ER33704,

ann.wrk.hrs\_hd\_2003 = ER24080, ann.wrk.hrs\_wf\_2003 = ER24091,

yearswrk\_hd\_2003 = ER23476, yearswrk.ft\_hd\_2003 = ER23477, yearswrk\_wf\_2003 = ER23384, yearswrk.ft\_wf\_2003 = ER23385,

int.num\_2004 = ER33801, seq.num\_2004 = ER33802, rel.head\_2004 = ER33803, yearswrk\_hd\_2004 = NA, yearswrk.ft\_hd\_2004 = NA,

yearswrk\_wf\_2004 = NA, yearswrk.ft\_wf\_2004 = NA, ann.wrk.hrs\_hd\_2004 = NA, ann.wrk.hrs\_wf\_2004 = NA,

employed\_hd\_2004 = ER27711D2, employed\_wf\_2004 = ER27711J5, wkswrk\_hd\_2004 = ER27711D3, wkswrk\_wf\_2004 = ER27711J6,

hrs.per.wk\_hd\_2004 = ER27711E8, hrs.per.wk\_wf\_2004 = ER27711L2,

int.num\_2005 = ER33801, seq.num\_2005 = ER33802, rel.head\_2005 = ER33803, age\_2005 = ER33804,

ann.wrk.hrs\_hd\_2005 = ER27886, ann.wrk.hrs\_wf\_2005 = ER27897,

yearswrk\_hd\_2005 = ER27444, yearswrk.ft\_hd\_2005 = ER27445, yearswrk\_wf\_2005 = ER27348, yearswrk.ft\_wf\_2005 = ER27349,

int.num\_2006 = ER33901, seq.num\_2006 = ER33902, rel.head\_2006 = ER33903, yearswrk\_hd\_2006 = NA, yearswrk.ft\_hd\_2006 = NA,

yearswrk\_wf\_2006 = NA, yearswrk.ft\_wf\_2006 = NA, ann.wrk.hrs\_hd\_2006 = NA, ann.wrk.hrs\_wf\_2006 = NA,

employed\_hd\_2006 = ER40686D2, employed\_wf\_2006 = ER40686J5, wkswrk\_hd\_2006 = ER40686D3, wkswrk\_wf\_2006 = ER40686J6,

hrs.per.wk\_hd\_2006 = ER40686E8, hrs.per.wk\_wf\_2006 = ER40686L2,

int.num\_2007 = ER33901, seq.num\_2007 = ER33902, rel.head\_2007 = ER33903, age\_2007 = ER33904,

ann.wrk.hrs\_hd\_2007 = ER40876, ann.wrk.hrs\_wf\_2007 = ER40887,

yearswrk\_hd\_2007 = ER40616, yearswrk.ft\_hd\_2007 = ER40617, yearswrk\_wf\_2007 = ER40523, yearswrk.ft\_wf\_2007 = ER40524,

int.num\_2008 = ER34001, seq.num\_2008 = ER34002, rel.head\_2008 = ER34003, yearswrk\_hd\_2008 = NA, yearswrk.ft\_hd\_2008 = NA,

yearswrk\_wf\_2008 = NA, yearswrk.ft\_wf\_2008 = NA, ann.wrk.hrs\_hd\_2008 = NA, ann.wrk.hrs\_wf\_2008 = NA,

employed\_hd\_2008 = ER46669, employed\_wf\_2008 = ER46680, wkswrk\_hd\_2008 = ER46670, wkswrk\_wf\_2008 = ER46681,

hrs.per.wk\_hd\_2008 = ER46671, hrs.per.wk\_wf\_2008 = ER46682,

int.num\_2009 = ER34001, seq.num\_2009 = ER34002, rel.head\_2009 = ER34003, age\_2009 = ER34004,

ann.wrk.hrs\_hd\_2009 = ER46767, ann.wrk.hrs\_wf\_2009 = ER46788,

yearswrk\_hd\_2009 = ER46594, yearswrk.ft\_hd\_2009 = ER46595, yearswrk\_wf\_2009 = ER46500, yearswrk.ft\_wf\_2009 = ER46501,

int.num\_2010 = ER34101, seq.num\_2010 = ER34102, rel.head\_2010 = ER34103, yearswrk\_hd\_2010 = NA, yearswrk.ft\_hd\_2010 = NA,

yearswrk\_wf\_2010 = NA, yearswrk.ft\_wf\_2010 = NA, ann.wrk.hrs\_hd\_2010 = NA, ann.wrk.hrs\_wf\_2010 = NA,

employed\_hd\_2010 = ER52070, employed\_wf\_2010 = ER52081, wkswrk\_hd\_2010 = ER52071, wkswrk\_wf\_2010 = ER52082,

hrs.per.wk\_hd\_2010 = ER52072, hrs.per.wk\_wf\_2010 = ER52083,

int.num\_2011 = ER34101, seq.num\_2011 = ER34102, rel.head\_2011 = ER34103, age\_2011 = ER34104,

ann.wrk.hrs\_hd\_2011 = ER52175, ann.wrk.hrs\_wf\_2011 = ER52196,

yearswrk\_hd\_2011 = ER51955, yearswrk.ft\_hd\_2011 = ER51956, yearswrk\_wf\_2011 = ER51861, yearswrk.ft\_wf\_2011 = ER51862,

int.num\_2012 = ER34201, seq.num\_2012 = ER34202, rel.head\_2012 = ER34203, yearswrk\_hd\_2012 = NA, yearswrk.ft\_hd\_2012 = NA,

yearswrk\_wf\_2012 = NA, yearswrk.ft\_wf\_2012 = NA, ann.wrk.hrs\_hd\_2012 = NA, ann.wrk.hrs\_wf\_2012 = NA,

employed\_hd\_2012 = ER57824, employed\_wf\_2012 = ER57872, wkswrk\_hd\_2012 = ER57825, wkswrk\_wf\_2012 = ER57873,

hrs.per.wk\_hd\_2012 = ER57839, hrs.per.wk\_wf\_2012 = ER57887,

int.num\_2013 = ER34201, seq.num\_2013 = ER34202, rel.head\_2013 = ER34203, age\_2013 = ER34204,

ann.wrk.hrs\_hd\_2013 = ER57976, ann.wrk.hrs\_wf\_2013 = ER57997,

yearswrk\_hd\_2013 = ER57711, yearswrk.ft\_hd\_2013 = ER57712, yearswrk\_wf\_2013 = ER57601, yearswrk.ft\_wf\_2013 = ER57602,

int.num\_2014 = ER34301, seq.num\_2014 = ER34302, rel.head\_2014 = ER34303, yearswrk\_hd\_2014 = NA, yearswrk.ft\_hd\_2014 = NA,

yearswrk\_wf\_2014 = NA, yearswrk.ft\_wf\_2014 = NA, ann.wrk.hrs\_hd\_2014 = NA, ann.wrk.hrs\_wf\_2014 = NA,

employed\_hd\_2014 = ER65004, employed\_wf\_2014 = ER65052, wkswrk\_hd\_2014 = ER65005, wkswrk\_wf\_2014 = ER65053,

hrs.per.wk\_hd\_2014 = ER65019, hrs.per.wk\_wf\_2014 = ER65067,

int.num\_2015 = ER34301, seq.num\_2015 = ER34302, rel.head\_2015 = ER34303, age\_2015 = ER34305,

ann.wrk.hrs\_hd\_2015 = ER65156, ann.wrk.hrs\_wf\_2015 = ER65177,

yearswrk\_hd\_2015 = ER64871, yearswrk.ft\_hd\_2015 = ER64872, yearswrk\_wf\_2015 = ER64732, yearswrk.ft\_wf\_2015 = ER64733,

int.num\_2016 = ER34501, seq.num\_2016 = ER34502, rel.head\_2016 = ER34503, yearswrk\_hd\_2016 = NA, yearswrk.ft\_hd\_2016 = NA,

yearswrk\_wf\_2016 = NA, yearswrk.ft\_wf\_2016 = NA, ann.wrk.hrs\_hd\_2016 = NA, ann.wrk.hrs\_wf\_2016 = NA,

employed\_hd\_2016 = ER71096, employed\_wf\_2016 = ER71144, wkswrk\_hd\_2016 = ER71097, wkswrk\_wf\_2016 = ER71145,

hrs.per.wk\_hd\_2016 = ER71111, hrs.per.wk\_wf\_2016 = ER71159,

int.num\_2017 = ER34501, seq.num\_2017 = ER34502, rel.head\_2017 = ER34503, age\_2017 = ER34504,

ann.wrk.hrs\_hd\_2017 = ER71233, ann.wrk.hrs\_wf\_2017 = ER71254,

yearswrk\_hd\_2017 = ER70943, yearswrk.ft\_hd\_2017 = ER70944, yearswrk\_wf\_2017 = ER70805, yearswrk.ft\_wf\_2017 = ER70806,

int.num\_2018 = ER34701, seq.num\_2018 = ER34702, rel.head\_2018 = ER34703, yearswrk\_hd\_2018 = NA, yearswrk.ft\_hd\_2018 = NA,

yearswrk\_wf\_2018 = NA, yearswrk.ft\_wf\_2018 = NA, ann.wrk.hrs\_hd\_2018 = NA, ann.wrk.hrs\_wf\_2018 = NA,

employed\_hd\_2018 = ER77118, employed\_wf\_2018 = ER77166, wkswrk\_hd\_2018 = ER77119, wkswrk\_wf\_2018 = ER77167,

hrs.per.wk\_hd\_2018 = ER77133, hrs.per.wk\_wf\_2018 = ER77181,

int.num\_2019 = ER34701, seq.num\_2019 = ER34702, rel.head\_2019 = ER34703, age\_2019 = ER34704,

ann.wrk.hrs\_hd\_2019 = ER77255, ann.wrk.hrs\_wf\_2019 = ER77276,

yearswrk\_hd\_2019 = ER76961, yearswrk.ft\_hd\_2019 = ER76962, yearswrk\_wf\_2019 = ER76816, yearswrk.ft\_wf\_2019 = ER76817

) %>%

# Turning to long format where key = varname for all vars except

# sex, 1968 household and person numbers

gather(key, value, -c(pernum68, intnum68, female)) %>%

# Create individual id, year column, removing year from variable label

mutate(indiv.id = paste(intnum68, pernum68, sep = "\_"),

year = as.numeric(str\_match(key, "[0-9]+")),

var = str\_remove(key, "\_[0-9]+[0-9]+")) %>%

dplyr::select(-key) %>%

# Grouping by individual id

group\_by(indiv.id) %>%

# Turning data back to wide format, each record is a person-year

spread(var, value, convert = T) %>%

mutate(rel.head = ifelse(rel.head %in% c(1, 10), "head", ifelse(rel.head %in% c(2, 20, 22), "wife", "other")),

wkswrktot = ifelse(wkswrktot > 52, NA, wkswrktot), # Recoding missing values as missing

wkspermonth = ifelse(wkspermonth > 112, NA, wkspermonth), # Recoding missing values as missing

monthswrk = ifelse(monthswrk > 12, NA, monthswrk)) %>% # Recoding missing values as missing

# Selecting only the individuals in our sample

filter(indiv.id %in% unique(psid\_obs$indiv.id))

# Labelling gap years for annual hours worked predictions: two separate

# labels because for eeach group of years, we use a different set of available

# variables to generate annual hours worked in the gap year predictions

gapyr.initial <- c(2000, 1998)

gapyr.later <- c(2002, 2004, 2006, 2008, 2010, 2012, 2014, 2016, 2018)

# Creating a new object cleaning the experience variables

psid\_exp.clean <-psid\_exp %>%

mutate(

age = na\_if(age, 999),

# Assigning variables corresponding to heads to Rs who are heads, same with wives

# Also setting missing codes to missing

#age = ifelse(rel.head == "head", age\_hd, age\_wf), age = ifelse(age >= 99, NA, age),

yrswrk = ifelse(rel.head == "head", yearswrk\_hd, yearswrk\_wf),

yrswrk.clean = ifelse(yrswrk > 97, NA, yrswrk),

yrswrk.ft = ifelse(rel.head == "head", yearswrk.ft\_hd, yearswrk.ft\_wf), yrswrk.ft.clean = ifelse(yrswrk.ft > 97, NA, yrswrk.ft),

# Following Blau & Khan procedure to censor measured years of experience to max out at years worked after age 18

yrswrk.clean.cen = ifelse(yrswrk.clean > age-18, age-18, yrswrk.clean),

yrswrk.ft.clean.cen = ifelse(yrswrk.ft.clean > age-18, age-18, yrswrk.ft.clean),

yrswrk.clean.cen = ifelse(yrswrk.clean.cen < 0, 0, yrswrk.clean.cen),

yrswrk.ft.clean.cen = ifelse(yrswrk.ft.clean.cen < 0, 0, yrswrk.ft.clean.cen),

# Cleaning employed, hours per week and weeks per year later gap year vars

wkswrk = ifelse(rel.head == "head", wkswrk\_hd, wkswrk\_wf),

wkswrk.clean = ifelse(wkswrk > 52, NA, wkswrk), hrswk = ifelse(rel.head == "head", hrs.per.wk\_hd, hrs.per.wk\_wf),

hrswk.clean = ifelse(hrswk >= 998, NA, hrswk), employed = ifelse(rel.head == "head", employed\_hd, employed\_wf),

ann.wrk.hrs = ifelse(rel.head == "head", ann.wrk.hrs\_hd, ann.wrk.hrs\_wf),

# Estimating annual work hours for the gap years- for the first set of gap years (1998, 2000)

# If wkswrktot non-missing, do wkswrktot\*wkspermonth; if wkswrktot is missing, do monthswrk\*4.33\*wkspermonth.

# For the remaining gap years: after cleaning, if coded as employed, do wkswrk \* hrs.per.wk

ann.wrk.hrs.pred = ifelse(year %in% gapyr.initial & is.na(wkswrktot), monthswrk\*4.33\*wkspermonth, ifelse(

year %in% gapyr.initial, wkswrktot\*wkspermonth, ifelse(

year %in% gapyr.later & employed == 1, wkswrk.clean\*hrswk.clean, ifelse(

year %in% gapyr.later, 0, ann.wrk.hrs))))) %>%

# Selecting relevant variables

dplyr::select(indiv.id, intnum68, pernum68, year, female, rel.head, age, ann.wrk.hrs,

ann.wrk.hrs.pred, yrswrk.clean.cen, yrswrk.ft.clean.cen) %>%

arrange(indiv.id, year) %>%

mutate(

# Filling in age variable in the gap year

age = ifelse(is.na(age) & lag(age) != 0, lag(age) + 1, age),

# The gap year fill variables take the measured years of experience variable as of the year

# prior to the gap year and assigns that value to the gap year. Reminder that these variables

# are only updated in the PSID in 1985 or once R joins/establishes a new household

yrswrk.gapyrfill = ifelse(year %in% c(gapyr.initial, gapyr.later), lag(yrswrk.clean.cen), yrswrk.clean.cen),

yrswrk.ft.gapyrfill = ifelse(year %in% c(gapyr.initial, gapyr.later), lag(yrswrk.ft.clean.cen), yrswrk.ft.clean.cen),

# Creating measures of working positive hours, working full-time hours, or working part-time hours

# based on the annual hours worked variable created w/ information on the gap years

wrk.pos = ifelse(ann.wrk.hrs.pred > 0, 1, 0), wrk.ft = ifelse(ann.wrk.hrs.pred >= 1500, 1, 0),

wrk.pt = ifelse(wrk.pos == 1 & wrk.ft != 1, 1, 0)) %>%

# Selecting relevant variables

dplyr::select(indiv.id, intnum68, pernum68, year, female, age, rel.head, ann.wrk.hrs, ann.wrk.hrs.pred,

yrswrk.gapyrfill, yrswrk.ft.gapyrfill, wrk.pos, wrk.ft, wrk.pt) %>%

ungroup() %>%

mutate(rowid = row\_number(indiv.id)) # Creating a measure of number of observations by individual

# This creates a new object that uses cumulative sums to generate the final experience measures for R by year

psid\_exp.clean.2 <- psid\_exp.clean %>%

mutate(

# The next two variable takes the observed value for yrswrk or yrswrk.ft the FIRST time a new value is

# observed for a respondnet. The PSID asks respondents about their years of total and full-time experience

# when they first join the survey or when they become new heads. They also ask this question of all Rs in 1985:

# These "fill" variables set any variable subsequent to a newly observed vaue as missing: so if a respondent

# has observed values for years of experience in 1978, 1985, and 2000, these variables capture the measured

# years of experience only in these "updated" years for that respondent, setting the other years for that R as NA

yrswrk\_fill = ifelse(rowid %in% distinct(psid\_exp.clean, indiv.id, yrswrk.gapyrfill, .keep\_all = T)$rowid,

yrswrk.gapyrfill, NA),

yrswrk.ft\_fill = ifelse(rowid %in% distinct(psid\_exp.clean, indiv.id, yrswrk.ft.gapyrfill, .keep\_all = T)$rowid,

yrswrk.ft.gapyrfill, NA)) %>%

dplyr::select(-rowid) %>% # Removes the row id variable

mutate(

# These next two variables uses the variables above as a baseline,

# which have missing values for that year UNLESS new information is collected on R's work experience in that year.

# If no new information is collected on R's experience that year, the variables below are filled as follows:

# if the respondent reports working positive hours/ft hours that year, the value for that year/variable takes a value of 1.

# If R is coded as NOT having worked positive/ft hours that year, the value for that year/variable takes a value of 0

yrswrk\_final = ifelse(is.na(yrswrk\_fill) & wrk.pos == 1, 1, ifelse(is.na(yrswrk\_fill) & wrk.pos == 0, 0, yrswrk\_fill)),

yrswrk.ft\_final = ifelse(is.na(yrswrk.ft\_fill) & wrk.ft == 1, 1, ifelse(is.na(yrswrk.ft\_fill) & wrk.ft == 0, 0, yrswrk.ft\_fill))) %>%

# This groups respondent by individual and by the "group" of years for that R with matching work experience info,

# meaning that no new work experience info was collected. The following variables cumulatively sum the variables

# that record new information on work experience, if recorded, and if not, count whether the respondent was

# working that year- this gives us our year-by-year measure of work experience- this measure is defined as the

# years of work experience LAST reported by R prior to that year PLUS the subsequent years in which R worked

# positive/ft hours up to the baseline year. For example, in 1990, if the individual's work experience was last

# updated in 1985, these variables will measure work experience as of 1985 + additional years of work experience

# for each year up to 1990, defined by the observed number of hours worked in those years

group\_by(indiv.id, yrswrk.gapyrfill) %>%

mutate(expt = cumsum(yrswrk\_final),

expf = cumsum(yrswrk.ft\_final))

# This final object restricts the object above to our target outcome and covariate years and merges

# the experience data to the rset of our outcome/covariate data.

benchmark.exp <- psid\_exp.clean.2 %>%

# We select 1979 and 2016 so that the experience data reflects experience earned

# as of the year prior to when our other variables are observed, then add one to merge in with the data

filter(year %in% c(1980, 1981, 1990, 1991, 1999, 2001, 2009, 2011, 2015, 2017, 2019)) %>%

dplyr::select(-c(ann.wrk.hrs, female, rel.head)) %>% # Remove variables already in the covariate data

# Merge in to the covariate data by individual id and year

left\_join(psid\_obs, ., by = c("indiv.id", "year")) %>%

mutate(expf = ifelse(expf > age.y-17 & age.y !=0, age.y-17, expf),

expt = ifelse(expt > age.y-17 & age.y !=0, age.y-17, expt),

age = age.x) %>%

dplyr::select(-c(age.x, age.y))

rm(psid\_exp, psid\_exp.clean, psid\_exp.clean.2)

# Generating fertility variables using the fertility history data

# Fertility history data has an id for each child

psid\_fert <- read\_xlsx("Raw Data/psid\_fertility/J295915.xlsx") %>%

# Selecting relevant columns

dplyr::select(intnum68 = CAH3, pernum68 = CAH4, record.type = CAH2, chnum1 = CAH10,

chnum2 = CAH11, yr.child.born = CAH15, total.children = CAH106) %>%

mutate(

bio.record = ifelse(record.type == 1, 1, 0), # Records whether child is biological or not

yr.child.born = ifelse(yr.child.born %in% c(9998, 9999), NA, yr.child.born), # Assings NAs to missing codes for year in which child was born

indiv.id = paste(intnum68, pernum68, sep = "\_"), # R id

child.id = paste(chnum1, chnum2, sep = "."), # Unique id for each child

total.children = ifelse(total.children == 98, NA, total.children)) %>% # Total number of chidren as of date last collected

# The row below expands the data to create a row for each individual id for each year

# in the range of years in which children were born- the child id column has the child id in

# the year in which that child was born to that individual. For other years the child id variable is set to zero

tidyr::complete(indiv.id, yr.child.born, fill = list(child.id = "0.0")) %>%

group\_by(indiv.id) %>%

arrange(indiv.id, yr.child.born) %>%

mutate(counter = row\_number(indiv.id)) %>%

mutate(

number = ifelse(child.id != "0.0", 1, 0), # This variable labels whether a child was born in that year

counter = cumsum(number), # This cumulates the number of children by year for each individual

# This variable documents the cumulative number of children by each year: if the cumulative number of children

# is greater than the measured total number of children, it takes the value of the total number of children

num.children = ifelse(total.children < counter & complete.cases(total.children), total.children, counter),

year = yr.child.born) %>%

dplyr::select(-number) %>%

dplyr::select(indiv.id, year, num.children)

benchmark.fert <- psid\_fert %>%

filter(year %in% c(1979, 1980, 1989, 1990, 1998, 2000, 2008, 2010, 2014, 2016, 2018)) %>% # Selecting years prior to when other vars measured

mutate(year = year + 1) %>%

# Joining with the covariate data

left\_join(benchmark.exp, ., by = c("indiv.id", "year")) %>%

# Creating variable: if age at first birth missing in the covariate data and R does not have

# any children as per the PSID fertility files, then age at first birth is set to 9999

mutate(afb = ifelse(is.na(age.first.birth) & num.children == 0, 9999, age.first.birth)) %>%

# Note: if R has more than one child born in a target year, then R will have more than one record

# for the target year in the fertility data and will thus have two rows in the merged data.

# In the procedure below, we keep only the last row for a given R in the merged data, which

# counts all children born in the target year if more than one child was born

group\_by(year, indiv.id) %>%

# First we count the total number of person-year observations for each R-year combination

# and for each PY record, we label whether it is the first record, second, or so on: the last

# record has the total number of children born as of that year, if more than one child was born

mutate(n = n(), rownum = row\_number()) %>%

ungroup() %>%

# The following variable labels the last row number of the total PY-observations observed for that person

# as the row to keep: this keeps the row that captures the total number of respondents born to that person

# as of that year, and drops the extra rows which capture if more than one child was born that year

mutate(keep = ifelse(rownum == n, "keep", "drop")) %>%

filter(keep == "keep") %>%

dplyr::select(-c(keep, rownum, n))

# PSID Fertility History files only started being collected in 1985. As a result, there is a subset

# of respondents observed in the early years who have attrited by 1985 (473, ~9 % of Rs) and are not in the fertility

# history file. For these respondents, we use data collected on the total number of kids and the

# age of the oldest child born to the head (all respondents in later years are observed in Fertility History file)

psid.misskids <- read.dta("Raw Data/psid/psid\_wrk.dta") %>%

# Selecting relevant variables & renaming by year- 1980

transmute(family\_id = ER30001, person\_number = ER30002, female = ifelse(ER32000 == 2, 1, 0),

indiv.id = paste(family\_id, person\_number, sep = "\_"),

rel.head\_1980 = ifelse(ER30315 == 1, "head", ifelse(ER30315 == 2, "wife", "other")),

rel.head\_1981 = ifelse(ER30345 == 1, "head", ifelse(ER30345 == 2, "wife", "other")),

age\_1980 = na\_if(ER30316, 999), age\_1981 = na\_if(ER30346, 999),

# Getting total number of children of head age of oldest kid of head

totkids.head\_1980 = na\_if(V7368, 99), age.oldest.kid.head\_1980 = na\_if(V7365,99),

totkids.head\_1981 = na\_if(V8020, 99), age.oldest.kid.head\_1981 = na\_if(V8017,99)) %>%

gather(key, value, -c(family\_id, person\_number, indiv.id, female)) %>% # Turning to long format

# Creating year variable based on the covariate label

mutate(year = case\_when(grepl("\_1980", key) ~ 1980,

grepl("\_1981", key) ~ 1981),

var = str\_remove(key, "\_[0-9]+[0-9]+")) %>%

dplyr::select(-key) %>% # Removing extra key variable

# Grouping by individual id

group\_by(indiv.id) %>%

# Turning data back to wide format, each record is a person-year

spread(var, value, convert = T) %>%

filter(rel.head %in% c("head", "wife")) %>%

mutate(num.children.hd = totkids.head, # For heads and wives, assigns total # of children of head to heads & their wives

# Generating year born for head's oldest child by subtracting age at 1980 interview from 1980

yr.fb.hd = ifelse(num.children.hd == 0, NA, 1980-age.oldest.kid.head),

yr.born = 1980-age, # Getting year born for respondents, subtracting their age from 1980

# Creating a measure for head/spouse's age when head's first child was born by subtracting

# the year head/wife was born from the year in which head's first child was born

afb.synth = ifelse(num.children.hd == 0, 9999, yr.fb.hd - yr.born)) %>%

dplyr::select(indiv.id, year, num.children.hd, yr.fb.hd, yr.born, afb.synth,

age.oldest.kid.head)

# Merging this additional data to the main data by individual id and year

benchmark.fert.synth <- left\_join(benchmark.fert, psid.misskids,

by = c("indiv.id", "year")) %>%

mutate( # Creating additional variables:

# If number of children missing, assigns number of children born to head in 1980

# This leaves us with a missing rate of .5 percent for 1980

num.children.synth = ifelse(is.na(num.children), num.children.hd, num.children),

# Creating an indicator variable for whether number of children comes from this imputing procedure:

# of R's in 1980, 91.2 % (4910) have the number of kids assigned with fertility history files,

# 8.3 % (446) use the imputed value for head, and .5 % (27) remain missing

dummy.nkids.synth = ifelse(is.na(num.children) & !is.na(num.children.synth), 1, 0),

# If age at first birth missing in the data, assigns age when head's oldest child was born

afb.synth = ifelse(is.na(afb), afb.synth, afb),

# Some Rs with >0 children have a value of 999 for afb.synth- this wrongly indicates they have no children

# This is the case for individuals with missing data on AFB and get assigned AFB == 9999 because the head

# in their household in 1980 has had no kids. But according to reports in the fertility files, these individuals

# themselves have had children. We update the variable above by coding age at first birth as missing for

# individuals that have missing age at first birth, but report having >0 children according to the fertility files

afb.synth = ifelse(afb.synth == 9999 & complete.cases(num.children) & num.children > 0, NA, afb.synth),

# Creating an indicator variable for whether age at first birth comes from this imputing procedure:

# of R's in 1980, 88 % 4751 have the number of kids assigned with fertility history files,

# 9.9 % (533) use the value imputed based on age of oldest child of head, and 1.8 % (99) remain missing

dummy.afb.synth = ifelse(is.na(afb) & !is.na(afb.synth), 1, 0))

# We then use the fertility history files to create an alternative measure of first birth

# for individuals with missing data on age at first birth. This procedure uses the fertility

# history files by observing the first year in which a respondent transitions from having no

# children to having a child and taking the respondent's age in that year as the age of first birth

# We first create a vector of ids with missing data on our previous measure of age at first birth ("afb.synth")

# Recall that this measure uses the year when the respondent was born and the year in which a respondent's

# first child ws born from the family files. Given that individuals who attrit from the sample before 1985 do

# not have this information in the family files, we use the the reported age of the head's oldest child

# to generate a measure of age at first birth as R's age when the respondent's oldest child was born for those

# who attrit from the sample before 1985 (for details on this procedure refer to the code chunk above, lines 803-846)

ids.miss <- benchmark.fert.synth %>% filter(is.na(afb.synth)) %>% dplyr::select(indiv.id) %>% pull()

# This takes an object with a record for each individual id in each year for which a child could be born

test <- psid\_fert %>%

ungroup() %>%

# We filter out "missing year" records for individuals with no children recorded

mutate(drop = ifelse(is.na(year) & num.children == 0, "drop", "keep")) %>%

filter(drop == "keep") %>%

dplyr::select(-drop) %>%

# This creates a variable with labeled row numbers for the entire data (not row number by person id)

mutate(rownum = row\_number())

# We then create a new object that records the row number in which the number of children in the data changes

# The "lenghts" column captures the number of rows between a change in value in number of children

# So "lengths" 27, 6, and 3 mean that the value of number of children first changes after 27 rows, then changes

# again 6 rows after that, then again 3 rows after that, and so on

t <- rle(test$num.children)[1] %>%

as.data.frame() %>%

# The "pos" column cumulatively sums the length column to get the specific row numbers in the data at which

# the number of children changes: "lengths" 27, 6, and 3 indicate that the value for number of children change

# after 27 rows, then 6 rows after, then 3 rows after. By cumulatively summing "lengths", we get the specific

# row number at which the number of children changes (27+6 = 33, 33+3 = 36, and so on)

mutate(pos = cumsum(lengths))

# We then go back to the object that has a record for each individual in each year in which a child is born

# and select only the row numbers in this dataframe at which the number of children changes

test2 <- test %>%

filter(rownum %in% t$pos) %>%

# We then select rows where the number of children == 0. These rows represent the last year in which

# the respondent is reported to have no children. In other words, the respondent had their first child

# in the time span between the year when num.children == 0 and the subsequent year where num.children == 1

filter(num.children == 0) %>%

# We then create a variable that marks this year as the "year of first birth" for that respondent. This

# measure captures that the respondent had their first kid prior to the subsequent year

mutate(yr.firstchild.born.fert = year) %>%

dplyr::select(indiv.id, yr.firstchild.born.fert)

# We then begin the process of merging this data with the rest of the observed data

# First, we get the individual ids, year, and age columns from the observed data

psid.obs.id.yr.age <- psid\_obs %>% dplyr::select(indiv.id, year, age)

# We also create a vector of unique ids in the observed data

benchmark.ids <- as.vector(unique(psid\_obs$indiv.id))

# We then create a new object that uses the observed data and merges it with our

# synthetic data on year in which a child is first reported according to the fertility files

# First, we create a dataframe that creates an observation for each individual in our synthetic

# fertility data file above, for each of the covariate and wage years in our observed data

test3 <- crossing(indiv.id = test2$indiv.id,

year = c(1979, 1980, 1989, 1990, 1998, 2000, 2008, 2010, 2014, 2016, 2018)) %>%

mutate(year = year + 1) %>%

# We select only individuals in the synthetic fertility data file that are also in the observed data

# (due to sample restrictions like being a head/wife in the outcome year, being present in the outcome yr)

filter(indiv.id %in% benchmark.ids) %>%

# We then merge this data structure with our synthetic fertililty file: This gives us a

# data structure that has an observation for each person in our synthetic fertility file who is also in

# our observed data, for each covariate/wage year, and a variable that captures the last year in which

# that respondent reported zero children before transitioning to reporting one child

left\_join(., test2, by = "indiv.id") %>%

# We then merge this data with our id-year-age column subset of the observed data to

# generate a new variable that captures the respondent's age in the year in which they

# are last observed with no children before transitioning to having their first child

left\_join(., psid.obs.id.yr.age, by = c("indiv.id", "year")) %>%

mutate(afb.ferhist = yr.firstchild.born.fert - (year - age)) %>%

dplyr::select(-age) # We select out age as to not have it repeated when we merge this object to our observed data

# We then merge in this data object to our observed data

benchmark.fert.final <- benchmark.fert.synth %>%

left\_join(., test3, by = c("indiv.id", "year")) %>%

# We create our final measures of fertility below

mutate(

# If age at first birth missing in the data, assigns age when head's oldest child was born

# This variable takes on the value of our first "synthetic" measure of fertility (using age when

# head's oldest child was born for the early year) when available. If this "synthetic" measure is

# missing, this variable takes on the value of our second "synthetic" measure, based on the last year

# in which a respondent was observed as childless when transitioning to having one child based on their

# fertility history. If both values are missing, this value is missing

afb.final = case\_when(is.na(afb.synth) ~ afb.ferhist, complete.cases(afb.synth) ~ afb.synth),

# This variable repeats the procedure above, but omits the step where we assign age at first birth

# based on the respondent's age when the head's oldest child was born for the early covariate year

afb.nosynth = ifelse(is.na(afb), afb.ferhist, afb),

# We then create categorical variables for parity and age at first birth based on these vars

num.kids.trunc = ifelse(num.children.synth == 0, "0", ifelse(

num.children.synth == 1, "1", ifelse(

num.children.synth == 2, "2", ifelse(

num.children.synth >=3, "3plus", NA)))),

afb.cat = ifelse(afb.final == 9999, "nokids", ifelse(

num.kids.trunc == 0, "nokids", ifelse(

afb.final <= 20, "20minus", ifelse(

afb.final <=23, "21to23", ifelse(

afb.final <= 27, "23to27", "27plus"))))))

rm(psid\_fert, benchmark.fert, psid.misskids, benchmark.fert.synth, test, t, test2, test3)

# Generating marital history from the marital history files

# The marital history files contain one record for each individual in the PSID since 1985

# Each R has at least one record, and each R has a record for each of their marriages

psid\_mar <- read.dta("Raw Data/psid\_marital/psid\_marital.dta") %>%

transmute(

intnum68 = MH2, pernum68 = MH3, indiv.id = paste(intnum68, pernum68, sep="\_"), # Creating variables IDing R's

# Variable indicating the order of this marriage record for R- recoding 99 to 0 marriages, 98 as missing

marriage.order = ifelse(MH9 == 99, 0, ifelse(MH9 == 98, NA, MH9)),

# Creating variables for the marriage date

month.married = ifelse(MH10 %in% c(98,99), NA, str\_pad(MH5, width = 2, pad = "0")),

yr.married = ifelse(MH11 %in% c(9998, 9999), NA, MH11),

marriagedate = mdy(paste(month.married, "01", yr.married, sep = "/")),

# This variable indicates the status of this marriage as of the latest marital history report for this R

status.marriage = case\_when(MH12 == 9 ~ "never.married", MH12 == 7 ~ "other/bigamist",

MH12 == 5 ~ "separated", MH12 == 4 ~ "divorced",

MH12 == 3 ~ "widowed", MH12 == 1 ~ "still.married"), # Implicitly coding 8 as NA

# Variables coding the date of the marriage's dissolution, whether divorced/widowed or separated

# 98 or 99 indicate either an uknown date or whether the respondent was never married/marriage hadn't ended

# 21-22-23-24 are codes for "seeasons", which are recoded to a month within that season

month.w.d = ifelse(MH13 %in% c(98,99), NA, ifelse(MH13 == 21, 2, ifelse(MH13 == 22, 5, ifelse(MH13 == 23, 8, ifelse(

MH13 == 24, 11, str\_pad(MH13, width = 2, pad = "0")))))),

yr.w.d = ifelse(MH14 %in% c(9998, 9999), NA, MH14),

divwiddate = mdy(paste(month.w.d, "01", yr.w.d, sep = "/")),

month.sep = ifelse(MH15 %in% c(98,99), NA, ifelse(MH15 == 21, 2, ifelse(MH15 == 22, 5, ifelse(MH15 == 23, 8, ifelse(

MH15 == 24, 11, str\_pad(MH15, width = 2, pad = "0")))))),

yr.sep = ifelse(MH16 %in% c(9998, 9999), NA, MH16),

sepdate = mdy(paste(month.sep, "01", yr.sep, sep = "/")),

# This variable captures the year in which marital history reports were last collected for that individual

lastyr.resp.marriage = MH17,

# This variable reprsents the number of marriages R has reported: if num-marriages == 0, never married

num.marriages = na\_if(MH18, 98),

# The following procedure aims to capture the status of the marriage record for each respondent as of the

# end of the year prior to the year in our sample. The next two variables code an "end date" for the marriage- if the

# the record is for a respondent who never married, we code never married. If the record is for an intact marriage,

# we code this "last date" as the last day of our wage year. If the record is for a marriage that has been

# dissolved, it codes the date at which that marriage was dissolved

lastdate\_1980 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-1979"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_1981 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-1980"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_1990 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-1989"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_1991 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-1990"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_1999 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-1998"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2001 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2000"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2009 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2008"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2011 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2010"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2015 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2014"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2017 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2016"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

lastdate\_2019 = as\_date(ifelse(status.marriage == "never.married", NA, ifelse(

status.marriage %in% c("still.married", "other/bigamist"),

mdy("12-31-2018"), ifelse(

status.marriage %in% c("divorced", "widowed"), divwiddate, ifelse(

status.marriage == "separated", sepdate, NA))))),

# Here, we code the status of this marriage as of the last day in our wage year

# If the record is for a respondent who's never married, we code it as never married:

# if the marriage start date in that record happens after the last day of our wage year, we code this

# marriage as not yet married. If the marriage start date is before the end of our wage year and the "last date"

# variable above (indicating when the marriage was dissolved, if dissolved by last obs period) is before

# the end of our wage year, we code this marriage as the status of that marriage as of the last obs period:

# If the marriage start date is before the end of our wage yeear, but the dissolution date is after the end

# of our wage year, then we code this record as "still married" during our wage year

status.marriage\_1980 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-1979"), "notyet.married", ifelse(

lastdate\_1980 < mdy("12-31-1979"), status.marriage, ifelse(

lastdate\_1980 >= mdy("12-31-1979"), "still.married", NA)))),

status.marriage\_1981 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-1980"), "notyet.married", ifelse(

lastdate\_1981 < mdy("12-31-1980"), status.marriage, ifelse(

lastdate\_1981 >= mdy("12-31-1980"), "still.married", NA)))),

status.marriage\_1990 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-1989"), "notyet.married", ifelse(

lastdate\_1990 < mdy("12-31-1989"), status.marriage, ifelse(

lastdate\_1990 >= mdy("12-31-1989"), "still.married", NA)))),

status.marriage\_1991 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-1990"), "notyet.married", ifelse(

lastdate\_1991 < mdy("12-31-1990"), status.marriage, ifelse(

lastdate\_1991 >= mdy("12-31-1990"), "still.married", NA)))),

status.marriage\_1999 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-1998"), "notyet.married", ifelse(

lastdate\_1999 < mdy("12-31-1998"), status.marriage, ifelse(

lastdate\_1999 >= mdy("12-31-1998"), "still.married", NA)))),

status.marriage\_2001 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2000"), "notyet.married", ifelse(

lastdate\_2001 < mdy("12-31-2000"), status.marriage, ifelse(

lastdate\_2001 >= mdy("12-31-2000"), "still.married", NA)))),

status.marriage\_2009 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2008"), "notyet.married", ifelse(

lastdate\_2009 < mdy("12-31-2008"), status.marriage, ifelse(

lastdate\_2009 >= mdy("12-31-2008"), "still.married", NA)))),

status.marriage\_2011 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2010"), "notyet.married", ifelse(

lastdate\_2011 < mdy("12-31-2010"), status.marriage, ifelse(

lastdate\_2011 >= mdy("12-31-2010"), "still.married", NA)))),

status.marriage\_2015 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2014"), "notyet.married", ifelse(

lastdate\_2015 < mdy("12-31-2014"), status.marriage, ifelse(

lastdate\_2015 >= mdy("12-31-2014"), "still.married", NA)))),

status.marriage\_2017 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2016"), "notyet.married", ifelse(

lastdate\_2017 < mdy("12-31-2016"), status.marriage, ifelse(

lastdate\_2017 >= mdy("12-31-2016"), "still.married", NA)))),

status.marriage\_2019 = ifelse(status.marriage == "never.married", "never.married", ifelse(

marriagedate > mdy("12-31-2018"), "notyet.married", ifelse(

lastdate\_2019 < mdy("12-31-2018"), status.marriage, ifelse(

lastdate\_2019 >= mdy("12-31-2018"), "still.married", NA)))),

# There are some missing values for the status of the marriage that we can ascertain:

# For example, there are some records which have a reported status of divorce but no divorce date.

# If the last year in which marital history information is recorded precedes our wage year, then

# we know that the divorce in this marriage occurred prior to the wage year, and hence, that the

# status of this marriage as of our wage year is a divorce (assuming no repartnering)

# Note that we can only make this assumption for years starting in 1985 and beyond because marital history

# is only collected starting 1985: we are missing status at 1981 for ~5 pct of all collected marriage records

status.marriage\_1990 = ifelse(is.na(status.marriage\_1990) & lastyr.resp.marriage <= 1989,

status.marriage, status.marriage\_1990),

status.marriage\_1991 = ifelse(is.na(status.marriage\_1991) & lastyr.resp.marriage <= 1990,

status.marriage, status.marriage\_1991),

status.marriage\_1999 = ifelse(is.na(status.marriage\_1999) & lastyr.resp.marriage <= 1998,

status.marriage, status.marriage\_1999),

status.marriage\_2001 = ifelse(is.na(status.marriage\_2001) & lastyr.resp.marriage <= 2000,

status.marriage, status.marriage\_2001),

status.marriage\_2009 = ifelse(is.na(status.marriage\_2009) & lastyr.resp.marriage <= 2008,

status.marriage, status.marriage\_2009),

status.marriage\_2011 = ifelse(is.na(status.marriage\_2011) & lastyr.resp.marriage <= 2010,

status.marriage, status.marriage\_2011),

status.marriage\_2015 = ifelse(is.na(status.marriage\_2015) & lastyr.resp.marriage <= 2014,

status.marriage, status.marriage\_2015),

status.marriage\_2017 = ifelse(is.na(status.marriage\_2017) & lastyr.resp.marriage <= 2016,

status.marriage, status.marriage\_2017),

status.marriage\_2019 = ifelse(is.na(status.marriage\_2019) & lastyr.resp.marriage <= 2018,

status.marriage, status.marriage\_2019)) %>%

dplyr::select(-c(month.married, yr.married, month.w.d, yr.w.d, month.sep, yr.sep,

intnum68, pernum68)) %>%

# We then get the status of each individual's marriage by the target year

# by pasting together the status of each marriage by the target year for each R

group\_by(indiv.id) %>%

summarise(marstat\_1980 = paste(unique(status.marriage\_1980), collapse = ', '),

marstat\_1981 = paste(unique(status.marriage\_1981), collapse = ', '),

marstat\_1990 = paste(unique(status.marriage\_1990), collapse = ', '),

marstat\_1991 = paste(unique(status.marriage\_1991), collapse = ', '),

marstat\_1999 = paste(unique(status.marriage\_1999), collapse = ', '),

marstat\_2001 = paste(unique(status.marriage\_2001), collapse = ', '),

marstat\_2009 = paste(unique(status.marriage\_2009), collapse = ', '),

marstat\_2011 = paste(unique(status.marriage\_2011), collapse = ', '),

marstat\_2015 = paste(unique(status.marriage\_2015), collapse = ', '),

marstat\_2017 = paste(unique(status.marriage\_2017), collapse = ', '),

marstat\_2019 = paste(unique(status.marriage\_2019), collapse = ', ')) %>%

gather(key, value, -indiv.id) %>%

mutate(year = as.numeric(numextract(key))) %>%

dplyr::select(-key) %>%

# Using the status for each marriage by the target year for each individual, we create a variable

# that codes the individual's marital status at the end of the target year: if the individual's "value"

# is only never married or only not yet married, we code as unmarried: if any of the individual's

# marriages have started by and are intact at the end of the wage yeaer, we code as married

# all of the other statuses are coded accordingly

mutate(marstat = ifelse(value %in% c("notyet.married", "never.married"), "unmarried", ifelse(

grepl("still.married", value), "married", ifelse(

grepl("other/bigamist", value), "other/bigamist", ifelse(

grepl("divorced", value), "prev.married", ifelse(

grepl("separated", value), "prev.married", ifelse(

grepl("widowed", value), "prev.married", ifelse(

value %in% c("notyet.married, NA", "NA, notyet.married"), NA, NA)))))))) %>%

dplyr::select(-value)

# Joining marital status data with the rest of our observed data

benchmark.final <- left\_join(benchmark.fert.final, psid\_mar, by = c("indiv.id", "year")) %>%

mutate(

# Creating factor variable for age of youngest child

age.youngest.cat = ifelse(is.na(age.youngest), NA, ifelse(

age.youngest == 0, "nokids", ifelse(

age.youngest <= 5 , "zerotofive", "sixtoseventeen"))),

# If marital status is missing as per the marital history reports, we impute the marital status of head

marstat.synth = ifelse(is.na(marstat), marstat.hd, marstat),

# Indicator variable labeling whether marital status was gleaned from marital history report or head's status

dummy.marstat.synth = ifelse(is.na(marstat) & complete.cases(marstat.synth), 1, 0),

# Creating additional categorical variables (codes are different just in 1990)

ed.factor = case\_when(year == 1990 & yrs.ed.fam < 4 ~ "LessthanHS",

year == 1990 & yrs.ed.fam == 4 ~ "HighSchool",

year == 1990 & yrs.ed.fam <= 6 ~ "SomeCollege",

year == 1990 & yrs.ed.fam > 6 & complete.cases(avdeg) & avdeg == 1 ~ "AdvDeg",

year == 1990 & yrs.ed.fam > 6 & complete.cases(ba) & ba == 1 ~ "BA",

year != 1990 & yrs.ed.fam < 12 ~ "LessthanHS",

year != 1990 & yrs.ed.fam == 12 ~ "HighSchool",

year != 1990 & yrs.ed.fam > 12 & complete.cases(avdeg) & avdeg == 1 ~ "AdvDeg",

year != 1990 & yrs.ed.fam > 12 & complete.cases(ba) & ba == 1 ~ "BA",

year != 1990 & yrs.ed.fam > 12 ~ "SomeCollege"),

region = case\_when(region == 1 ~ "Northeast",

region == 2 ~ "Northcentral",

region == 3 ~ "South",

region == 4 ~ "West",

region == 5 ~ "Alaska.Hawaii",

region == 6 ~ "Foreign.Country")) %>%

# This variable recodes the 2010 occs to match to IPUMS occupational characteristics

# There are 28 OCC2010 codes that have no match in the IPUMS data- this depends on the year

# (10 missing occs in 1980 and 18 in 2017)

# Recoding these occupations to similar occupations in that year when missing

mutate(

occ2010.ipums = case\_when(

# Nursing, Psychiatric, and Home Health Aides (3600) recoded to

# Medical Assistants and Other Healthcare Support Occupations, nec (3650)

year %in% c(1980, 1981) & occ2010 == 3600 ~ 3650,

# Other Installation, Maintenance, and Repair Workers Including Wind Turbine Service

# Technicians, and Commercial Divers, and Signal and Track Switch Repairers (7630) recoded

# to Helpers--Installation, Maintenance, and Repair Workers (7610)

year %in% c(1980, 1981) & occ2010 == 7630 ~ 7610,

# Automotive and Watercraft Service Attendants (9360) recoded to Transportation workers NEC (9420)

year %in% c(1980, 1981) & occ2010 == 9360 ~ 9420,

# Professional, Research, or Technical Workers, nec (1980) recoded to

# Life, Physical, and Social Science Technicians, nec (1960)

occ2010 == 1980 ~ 1960,

# Food Service and Lodging Managers (310) recoded to Managers, nec (including Postmasters) (430)

year %in% c(1980, 1981) & occ2010 == 310 ~ 430,

# Dishwashers (4140) recoded to Food preparation and serving related workers, nec (4130)

year %in% c(1980, 1981) & occ2010 == 4140 ~ 4130,

# Avionic Technicians (7030) and Electronic Home Entertainment Equipment Installers and Repairers (7120)

# recoded to Electronic Repairs, nec (7125)

year %in% c(1980, 1981) & occ2010 == 7030 | occ2010 == 7120 ~ 7125,

# Chiropractors (3000) recoded to Therapists nec (3240)

year %in% c(1980, 1981) & occ2010 == 3000 ~ 3240,

# Tool Grinders, Filers, and Sharpeners (8210) recoded to Metal workers and plastic workers, nec (8220)

year %in% c(2015, 2017, 2019) & occ2010 == 8210 ~ 8220,

# Dredge, Excavating, and Loading Machine Operators (9520) recoded to Material moving workers, nec (9750)

year %in% c(2015, 2017, 2019) & occ2010 == 9520 ~ 9750,

# Manufactured Building and Mobile Home Installers (7550) recoded to

# Helpers--Installation, Maintenance, and Repair Workers (7610)

year %in% c(2015, 2017, 2019) & occ2010 == 7550 ~ 7610,

# Paving, Surfacing, and Tamping Equipment Operators (6300) recoded to

# Construction equipment operators except paving, surfacing, and tamping equipment operators (6320)

year %in% c(2015, 2017, 2019) & occ2010 == 6300 ~ 6320,

# Shoe machine operators and tenders (8340) recoded to Shoe and Leather Workers and Repairers (8330)

year %in% c(2015, 2017, 2019) & occ2010 == 8340 ~ 8330,

# Electronic Equipment Installers and Repairers, Motor Vehicles (7110) recoded to Electronic Repairs, nec (7125)

year %in% c(2015, 2017, 2019) & occ2010 == 7110 ~ 7125,

# Paperhangers (6430) and Reinforcing Iron and Rebar Workers (6500) recoded to Construction workers nec (6765)

year %in% c(2015, 2017, 2019) & occ2010 %in% c(6430, 6500) ~ 6765,

# Computer operators (5800) recoded to Officers and Administrative support nec (5940)

year %in% c(2015, 2017, 2019) & occ2010 == 5800 ~ 5940,

# Aircraft Structure, Surfaces, Rigging, and Systems Assemblers (7710) recoded to

# Assemblers and Fabricators, nec (7750)

year %in% c(2015, 2017, 2019) & occ2010 == 7710 ~ 7750,

# Textile Knitting and Weaving Machine Setters, Operators, and Tenders (8410) recoded to

# Textile, Apparel, and Furnishings workers, nec (8460)

year %in% c(2015, 2017, 2019) & occ2010 == 8410 ~ 8460,

# Funeral Directors (320) recoded to Managers nec (430)

year %in% c(2015, 2017, 2019) & occ2010 == 320 ~ 430,

# Brokerage Clerks (5200) recoded to Information and Record Clerks, All Other (5420)

year %in% c(2015, 2017, 2019) & occ2010 == 5200 ~ 5420,

# First-Line Supervisors of Gaming Workers (4300) recoded to Personal Care and Service Workers, All Other (4650)

year %in% c(2015, 2017, 2019) & occ2010 == 4300 ~ 4650,

# Geological and Petroleum Technicians, and Nuclear Technicians (1930) recoded to

# Life, Physical, and Social Science Technicians, nec (1960)

year %in% c(2015, 2017, 2019) & occ2010 == 1930 ~ 1960,

# Plating and Coating Machine Setters, Operators, and Tenders, Metal and Plastic (8200) recoded to

# Metal workers and plastic workers, nec (8220)

year %in% c(2015, 2017, 2019) & occ2010 == 8200 ~ 8220,

# Counter Attendant, Cafeteria, Food Concession, and Coffee Shop (4060) recoded to

# Food preparation and serving related workers, nec (4130)

year %in% c(2015, 2017, 2019) & occ2010 == 4060 ~ 4130,

# Cleaning, Washing, and Metal Pickling Equipment Operators and Tenders (8860) recoded to

# Other production workers including semiconductor processors and cooling and freezing equipment operators (8965)

year %in% c(2015, 2017, 2019) & occ2010 == 8860 ~ 8965,

TRUE ~ 00000),

occ2010.ipums = ifelse(occ2010.ipums == 00000, occ2010, occ2010.ipums)) %>%

dplyr::select(indiv.id, year, female, family\_id, person\_number, samp\_error\_stratum, samp\_error\_cluster,

samp.inc.1981, samp.inc.2017, imm.sample.97, imm.sample.17, imm.sample, latino.sample, missing.interview,

perwt, perwt.imm, perwt.lat, perwt.lat.imm, age, lead.age, rel.head, marstat.hd, numkids.fu, race, region, lead.region, yrs.ed.fam,

avdeg, ba, ann.wrk.hrs, lead.ann.wrk.hrs, wrk.pos, wrk.ft, wrk.pt, expt, expf, govt.job, govt.job\_alljobs, union,

self.emp, self.emp\_alljobs, self.emp.synth, ind.orig, ind1990, ind.2d, occ.orig, occ2010, wages, lead.wage, wkswrk,

lead.wkswrk, emp.tenure, hrs.wrk.wk, lead.hrs.wrk.wk, military, agriculture, num.children,

num.children.hd, num.children.synth, dummy.nkids.synth, num.kids.trunc, age.first.birth, afb,

afb.synth, dummy.afb.synth, afb.ferhist, afb.final, afb.nosynth, afb.cat, marstat, marstat.hd,

marstat.synth, dummy.marstat.synth, ed.factor,occ2010.ipums, housework,

age.youngest, age.youngest.cat, faminc)

# Loading data from IPUMS files to merge in occupational characteristics

# NOTE: To load data, you must download both the extract's data and the DDI and also set the working directory to

# the folder with these files (or change the path below).

if (!require("ipumsr")) stop(

"Reading IPUMS data into R requires the ipumsr package.

It can be installed using the following command:

install.packages('ipumsr')")

# Reading in the data

ddi <- read\_ipums\_ddi("Raw Data/ipums\_occs/usa\_00054.xml")

ipums.occs.raw <- read\_ipums\_micro(ddi)

# Selecting the relevant data:

ipums.occs <- ipums.occs.raw %>%

mutate(female = ifelse(SEX ==2, 1, 0),

# Creating indicator for whether individual meets our hours/weeks worked ft criteria

wrk.ft = ifelse(WKSWORK2==6 & UHRSWORK>=35, 1, 0),

# Indicators for working part-time (non zero hours & weeks, non-FT)

wrk.pt = ifelse(WKSWORK2 > 0 & UHRSWORK > 0 & wrk.ft == 0, 1, 0),

wrk.ov = ifelse(wrk.ft == 1 & UHRSWORK >= 50, 1, 0), # overwork (FT, Weekly Hours >= 50)

INCADJ = as.numeric(INCWAGE \* CPI99 \* 1.471), # Adjusting income to same year as our PSID data

EDUC = as.numeric(EDUC),

logincadj = log(INCADJ)) %>%

filter(CLASSWKR==2) %>% # EXCLUDING THOSE NOT WORKING FOR WAGES (SELF-EMP)

filter(EMPSTAT ==1) %>% # INCLUDING ONLY THOSE WHO ARE EMPLOYED

filter(AGE >= 35 & AGE <= 45) # Defining occupational characteristics only for those in this decade

occ.pay <- ipums.occs %>%

# Defining average income of MEN working full-time in a given occupation during our wage years

filter(female == 0, wrk.ft == 1) %>%

group\_by(YEAR, OCC2010) %>%

summarise(occ.mean.inc = wtd.mean(INCADJ, w = PERWT))

ipums.data <- ipums.occs %>%

group\_by(YEAR, OCC2010) %>%

# Creating a measure of pct female in a given occupation during our wage years

summarise(occ.pct.female = wtd.mean(female, w = PERWT),

occ.mean.educ = wtd.mean(EDUC, w = PERWT),

occ.wrk.pt = wtd.mean(wrk.pt, w = PERWT),

occ.wrk.ov = wtd.mean(wrk.ov, w = PERWT)) %>%

filter(occ.pct.female < 1) %>%

left\_join(., occ.pay, by = c("YEAR", "OCC2010")) %>%

rename(year = "YEAR", occ2010.ipums = "OCC2010") %>%

ungroup() %>%

mutate(year = ifelse(year == 2019, 2015, ifelse(

year == 2000, 1999, ifelse(year == 2012, 2009, year))),

occ2010.ipums = as.numeric(occ2010.ipums))

# Creating the object to merge to our observed data

# Note: this object will create a row for each year in our observed data

# and each IPUMS occupation. Both early years will be assigned the average characteristics

# of the occupation in 1980, and both later years will be assigned the average characteristics

# of the occupation in the ACS 2013-2018 five year file. We merge this to our time-varying PSID occupation dummies

benchmark.ipums <- crossing(occ2010.ipums = ipums.data$occ2010.ipums,

year = c(1980, 1981, 1990, 1991, 1999, 2001, 2009, 2011, 2015, 2017, 2019)) %>%

left\_join(., ipums.data, by = c("occ2010.ipums", "year")) %>%

group\_by(occ2010.ipums) %>%

fill(starts\_with("occ."), .direction = "down") %>%

# Joining the IPUMS data with our PSID data by year

# Note that we merge the same occupation characteristics (1980 Census, and from the 2013-2018 ACS)

# to the different years in the same period- so 1980 census characteristics are matched to occupation dummies

# as measured in 1980 and 1981, and ACS 2013-2018 characteristics are matched into occupation dummies in 2015 and 2017

right\_join(., benchmark.final, by = c("year", "occ2010.ipums")) %>%

# Creating indicators for different kinds of sample restrictions:

# Excluding agriculture and military, excluding the self-employed, including only FT workers,

# excluding individuals based on wage restrictions (non-zero wages, above 4000 annual earnings)

mutate(

samp.exc\_mil.ag = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & military == 1 | agriculture == 1 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & ind.2d %in% c("Active.Duty.Military", "Agriculture") |

occ2010 %in% c(seq(9750, 9830), seq(5940, 6130)) ~ 1,

TRUE ~ 0),

samp.exc\_region = case\_when(

region %in% c("Alaska.Hawaii", "Foreign.Country") ~ 1,

TRUE ~ 0),

samp.exc\_selfemp = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & self.emp.synth == 1 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & self.emp == 1 ~ 1,

TRUE ~ 0),

samp.exc\_zerowage = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & lead.wage == 0 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & wages == 0 ~ 1,

TRUE ~ 0),

samp.exc\_wagethresh = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & lead.wage < 4000 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & wages < 4000 ~ 1,

TRUE ~ 0),

samp.inc\_ft = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & lead.wkswrk >= 26 & lead.hrs.wrk.wk >= 35 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & wkswrk >= 26 & hrs.wrk.wk >= 35 ~ 1,

TRUE ~ 0),

samp.inc\_age = case\_when(

year %in% c(1980, 1990, 1999, 2009, 2015, 2017) & lead.age >= 30 | lead.age <= 60 ~ 1,

year %!in% c(1980, 1990, 1999, 2009, 2015, 2017) & age >= 30 | age <= 60 ~ 1,

TRUE ~ 0),

samp.inc.final = ifelse(samp.exc\_mil.ag != 1 & samp.exc\_selfemp != 1 & samp.exc\_region != 1 &

samp.exc\_zerowage != 1 & samp.exc\_wagethresh != 1 &

samp.inc\_ft == 1 & samp.inc\_age == 1, 1, 0))

write\_csv(benchmark.ipums, "Output/Clean Data/psid\_clean\_2019.csv")

toc()