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

# PROJECT: KILLEWALD- GENDER WAGE DISTRIBUTION

# PSID WIDE IMPUTATION

# AUTHOR: NINO CRICCO

# LAST UPDATED: 09/06/2020 (dmy)

# RUNTIME: 1131.871 sec (~19 min)

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

library(tidyverse)

library(mice)

library(naniar)

library(gridExtra)

library(tictoc)

tic()

# Loading helper functions

source("Jobs/helperfunctions.R")

psid <- read\_csv("Output/Clean Data/psid\_clean\_2019.csv") %>%

# We set wages for both the main timing and the alternate timing

# to equal wages as measured in 1981, 2017

mutate(wages = ifelse(year %in% c(1980, #1990,

1999, 2009, 2015, 2017), lead.wage, wages),

ann.wrk.hrs = ifelse(year %in% c(1980, #1990,

1999, 2009, 2015, 2017), lead.ann.wrk.hrs, ann.wrk.hrs)) %>%

# Removing "lead" variables used to generate the sample restriction +

# annual work hours and wages for the alternate variable timings

dplyr::select(-c(starts\_with("lead"), family\_id, person\_number,

military, agriculture))

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

# IMPUTATION

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

# Creating a vector of variables that will later be converted to factors

factor.vars <- c("indiv.id", "samp\_error\_stratum", "samp\_error\_cluster",

"rel.head\_1980", "rel.head\_1981", "rel.head\_1990", "rel.head\_1991",

"rel.head\_1999", "rel.head\_2001", "rel.head\_2009", "rel.head\_2011",

"rel.head\_2015", "rel.head\_2017", "rel.head\_2019",

"marstat.hd\_1980", "marstat.hd\_1981", "marstat.hd\_1990", "marstat.hd\_1991",

"marstat.hd\_1999", "marstat.hd\_2001", "marstat.hd\_2009", "marstat.hd\_2011",

"marstat.hd\_2015", "marstat.hd\_2017", "marstat.hd\_2019",

"race\_1980", "race\_1981", "race\_1990", "race\_1991", "race\_1999", "race\_2001",

"race\_2009", "race\_2011", "race\_2015", "race\_2017", "race\_2019",

"region\_1980", "region\_1981", "region\_1990", "region\_1991", "region\_1999",

"region\_2001", "region\_2009", "region\_2011","region\_2015", "region\_2017", "region\_2019",

"ind.orig\_1980", "ind.orig\_1981", "ind.orig\_1990", "ind.orig\_1991", "ind.orig\_1999",

"ind.orig\_2001", "ind.orig\_2009", "ind.orig\_2011", "ind.orig\_2015", "ind.orig\_2017", "ind.orig\_2019",

"ind1990\_1980", "ind1990\_1981", "ind1990\_1990", "ind1990\_1991", "ind1990\_1999",

"ind1990\_2001", "ind1990\_2009", "ind1990\_2011", "ind1990\_2015", "ind1990\_2017", "ind1990\_2019",

"ind.2d\_1980", "ind.2d\_1981", "ind.2d\_1990", "ind.2d\_1991", "ind.2d\_1999",

"ind.2d\_2001", "ind.2d\_2009", "ind.2d\_2011", "ind.2d\_2015", "ind.2d\_2017", "ind.2d\_2019",

"occ.orig\_1980", "occ.orig\_1981", "occ.orig\_1990", "occ.orig\_1991", "occ.orig\_1999",

"occ.orig\_2001", "occ.orig\_2009", "occ.orig\_2011", "occ.orig\_2015", "occ.orig\_2017", "occ.orig\_2019",

"occ2010\_1980", "occ2010\_1981", "occ2010\_1990", "occ2010\_1991", "occ2010\_1999",

"occ2010\_2001", "occ2010\_2009", "occ2010\_2011", "occ2010\_2015", "occ2010\_2017", "occ2010\_2019",

"num.kids.trunc\_1980", "num.kids.trunc\_1981", "num.kids.trunc\_1990", "num.kids.trunc\_1991", "num.kids.trunc\_1999",

"num.kids.trunc\_2001", "num.kids.trunc\_2009", "num.kids.trunc\_2011", "num.kids.trunc\_2015", "num.kids.trunc\_2017", "num.kids.trunc\_2019",

"afb.cat\_1980", "afb.cat\_1981", "afb.cat\_1990", "afb.cat\_1991", "afb.cat\_1999",

"afb.cat\_2001", "afb.cat\_2009", "afb.cat\_2011", "afb.cat\_2015", "afb.cat\_2017", "afb.cat\_2019",

"age.youngest\_1980", "age.youngest\_1981", "age.youngest\_1990", "age.youngest\_1991", "age.youngest\_1999",

"age.youngest\_2001", "age.youngest\_2009", "age.youngest\_2011", "age.youngest\_2015", "age.youngest\_2017", "age.youngest\_2019",

"marstat\_1980", "marstat\_1981", "marstat\_1990", "marstat\_1991", "marstat\_1999",

"marstat\_2001", "marstat\_2009", "marstat\_2011", "marstat\_2015", "marstat\_2017", "marstat\_2019",

"marstat.synth\_1980", "marstat.synth\_1981", "marstat.synth\_1990", "marstat.synth\_1991", "marstat.synth\_1999",

"marstat.synth\_2001", "marstat.synth\_2009", "marstat.synth\_2011", "marstat.synth\_2015", "marstat.synth\_2017", "marstat.synth\_2019",

"ed.factor\_1980", "ed.factor\_1981", "ed.factor\_1990", "ed.factor\_1991", "ed.factor\_1999",

"ed.factor\_2001", "ed.factor\_2009", "ed.factor\_2011", "ed.factor\_2015", "ed.factor\_2017", "ed.factor\_2019")

# Transforming the data to wide format by period for imputation

# First doing it for the first period

period1 <- psid %>%

filter(year %in% c(1980, 1981)) %>% # Selecting two early years

# Transforming data to long format, except for id, year, and sample restriction vars

gather(variable, value, - c(indiv.id, year, female)) %>%

# Turnig data to wide format

pivot\_wider(id\_cols = c(indiv.id, female),

names\_from = c(variable, year), values\_from = value) %>%

# Transforming factors to factors, all other vars to numeric

mutate\_at(vars(one\_of(factor.vars)), as\_factor) %>%

mutate\_at(vars(-one\_of(factor.vars)), as.numeric)

# Initializing the imputation

init <- mice(period1, maxit = 0, method = "cart", seed = 500) # Set classiffication/regression tree as dflt method

meth <- init$method # Sets the method of imputation for each column

predM <- init$predictorMatrix # Establishes the predictor matrix

# Setting id and weight variables in the predictor matrix to zero so that they don't affect the imputation

predM[, c("indiv.id", "perwt\_1980", "perwt\_1981")]= 0

predM[, c("occ2010.ipums\_1980", "occ2010.ipums\_1981", "occ2010\_1980", "occ2010\_1981",

"occ.orig\_1980", "occ.orig\_1981", "ind.orig\_1980", "ind.orig\_1981", "ind1990\_1980", "ind1990\_1981",

"ed.factor\_1981", "ed.factor\_1980", "self.emp.synth\_1981", "self.emp.synth\_1981",

"num.kids.trunc\_1980", "num.kids.trunc\_1981", "dummy.afb.synth\_1980", "dummy.afb.synth\_1981",

"dummy.marstat.synth\_1980", "dummy.marstat.synth\_1981",

"dummy.nkids.synth\_1980", "dummy.nkids.synth\_1981")]= 0

# Running imputation with the specs outlined above

full.imp.p1 <- mice(period1, method = meth, predictorMatrix=predM, m = 5, seed = 500)

# Creating a long version of the imputed data

implong.p1 <- mice::complete(full.imp.p1, action = "long", include = TRUE) %>%

# Turnig data back to long format

gather(var, value, -c(.imp, .id, indiv.id, female)) %>%

separate(var, c("var", "year"), "\_19") %>%

mutate(year = case\_when(year == 80 ~ 1980,

year == 81 ~ 1981)) %>%

spread(var, value)

# Transforming the data to wide format by period for imputation

# First doing it for the first period

period2 <- psid %>%

filter(year %in% c(1990, 1991)) %>% # Selecting two early years

# Transforming data to long format, except for id, year, and sample restriction vars

gather(variable, value, - c(indiv.id, year, female)) %>%

# Turnig data to wide format

pivot\_wider(id\_cols = c(indiv.id, female),

names\_from = c(variable, year), values\_from = value) %>%

# Transforming factors to factors, all other vars to numeric

mutate\_at(vars(one\_of(factor.vars)), as\_factor) %>%

mutate\_at(vars(-one\_of(factor.vars)), as.numeric)

# Initializing the imputation

init <- mice(period2, maxit = 0, method = "cart", seed = 500) # Set classiffication/regression tree as dflt method

meth <- init$method # Sets the method of imputation for each column

predM <- init$predictorMatrix # Establishes the predictor matrix

# Setting id and weight variables in the predictor matrix to zero so that they don't affect the imputation

predM[, c("indiv.id", "perwt\_1990", "perwt\_1991")]= 0

predM[, c("occ2010.ipums\_1990", "occ2010.ipums\_1991", "occ2010\_1990", "occ2010\_1991",

"occ.orig\_1990", "occ.orig\_1991", "ind.orig\_1990", "ind.orig\_1991", "ind1990\_1990", "ind1990\_1991",

"ed.factor\_1991", "ed.factor\_1990", "self.emp.synth\_1991", "self.emp.synth\_1991",

"num.kids.trunc\_1990", "num.kids.trunc\_1991", "dummy.afb.synth\_1990", "dummy.afb.synth\_1991",

"dummy.marstat.synth\_1990", "dummy.marstat.synth\_1991",

"dummy.nkids.synth\_1990", "dummy.nkids.synth\_1991")]= 0

# Running imputation with the specs outlined above

full.imp.p2 <- mice(period2, method = meth, predictorMatrix=predM, m = 5, seed = 500)

# Creating a long version of the imputed data

implong.p2 <- mice::complete(full.imp.p2, action = "long", include = TRUE) %>%

# Turnig data back to long format

gather(var, value, -c(.imp, .id, indiv.id, female)) %>%

separate(var, c("var", "year"), "\_19") %>%

mutate(year = case\_when(year == 90 ~ 1990,

year == 91 ~ 1991)) %>%

spread(var, value)

# Transforming the data to wide format by period for imputation

# First doing it for the first period

period3 <- psid %>%

filter(year %in% c(1999, 2001)) %>% # Selecting two early years

# Transforming data to long format, except for id, year, and sample restriction vars

gather(variable, value, - c(indiv.id, year, female)) %>%

# Turnig data to wide format

pivot\_wider(id\_cols = c(indiv.id, female),

names\_from = c(variable, year), values\_from = value) %>%

# Transforming factors to factors, all other vars to numeric

mutate\_at(vars(one\_of(factor.vars)), as\_factor) %>%

mutate\_at(vars(-one\_of(factor.vars)), as.numeric)

# Initializing the imputation

init <- mice(period3, maxit = 0, method = "cart", seed = 500) # Set classiffication/regression tree as dflt method

meth <- init$method # Sets the method of imputation for each column

predM <- init$predictorMatrix # Establishes the predictor matrix

# Setting id and weight variables in the predictor matrix to zero so that they don't affect the imputation

predM[, c("indiv.id", "perwt\_1999", "perwt\_2001")]= 0

predM[, c("occ2010.ipums\_1999", "occ2010.ipums\_2001", "occ2010\_1999", "occ2010\_2001",

"occ.orig\_1999", "occ.orig\_2001", "ind.orig\_1999", "ind.orig\_2001", "ind1990\_1999", "ind1990\_2001",

"ed.factor\_2001", "ed.factor\_1999", "self.emp.synth\_2001", "self.emp.synth\_2001",

"num.kids.trunc\_1999", "num.kids.trunc\_2001", "dummy.afb.synth\_1999", "dummy.afb.synth\_2001",

"dummy.marstat.synth\_1999", "dummy.marstat.synth\_2001",

"dummy.nkids.synth\_1999", "dummy.nkids.synth\_2001")]= 0

# Running imputation with the specs outlined above

full.imp.p3 <- mice(period3, method = meth, predictorMatrix=predM, m = 5, seed = 500)

# Creating a long version of the imputed data

implong.p3 <- mice::complete(full.imp.p3, action = "long", include = TRUE) %>%

# Turnig data back to long format

gather(var, value, -c(.imp, .id, indiv.id, female)) %>%

separate(var, c("var", "year"), "\_\\d\\d") %>%

mutate(year = case\_when(year == "99" ~ 1999,

year == "01" ~ 2001)) %>%

spread(var, value)

# Transforming the data to wide format by period for imputation

# First doing it for the first period

period4 <- psid %>%

filter(year %in% c(2009, 2011)) %>% # Selecting two early years

# Transforming data to long format, except for id, year, and sample restriction vars

gather(variable, value, - c(indiv.id, year, female)) %>%

# Turnig data to wide format

pivot\_wider(id\_cols = c(indiv.id, female),

names\_from = c(variable, year), values\_from = value) %>%

# Transforming factors to factors, all other vars to numeric

mutate\_at(vars(one\_of(factor.vars)), as\_factor) %>%

mutate\_at(vars(-one\_of(factor.vars)), as.numeric)

# Initializing the imputation

init <- mice(period4, maxit = 0, method = "cart", seed = 500) # Set classiffication/regression tree as dflt method

meth <- init$method # Sets the method of imputation for each column

predM <- init$predictorMatrix # Establishes the predictor matrix

# Setting id and weight variables in the predictor matrix to zero so that they don't affect the imputation

predM[, c("indiv.id", "perwt\_2009", "perwt\_2011")]= 0

predM[, c("occ2010.ipums\_2009", "occ2010.ipums\_2011", "occ2010\_2009", "occ2010\_2011",

"occ.orig\_2009", "occ.orig\_2011", "ind.orig\_2009", "ind.orig\_2011", "ind1990\_2009", "ind1990\_2011",

"ed.factor\_2011", "ed.factor\_2009", "self.emp.synth\_2011", "self.emp.synth\_2011",

"num.kids.trunc\_2009", "num.kids.trunc\_2011", "dummy.afb.synth\_2009", "dummy.afb.synth\_2011",

"dummy.marstat.synth\_2009", "dummy.marstat.synth\_2011",

"dummy.nkids.synth\_2009", "dummy.nkids.synth\_2011")]= 0

# Running imputation with the specs outlined above

full.imp.p4 <- mice(period4, method = meth, predictorMatrix=predM, m = 5, seed = 500)

# Creating a long version of the imputed data

implong.p4 <- mice::complete(full.imp.p4, action = "long", include = TRUE) %>%

# Turnig data back to long format

gather(var, value, -c(.imp, .id, indiv.id, female)) %>%

separate(var, c("var", "year"), "\_20") %>%

mutate(year = case\_when(year == "09" ~ 2009,

year == "11" ~ 2011)) %>%

spread(var, value)

# Repeating the same procedure for the second period

period5 <- psid %>%

filter(year %in% c(2017, 2019)) %>% # Selecting two early years

# Transforming data to long format, except for id, year, and sample restriction vars

gather(variable, value, - c(indiv.id, year, female)) %>%

# Turnig data to wide format

pivot\_wider(id\_cols = c(indiv.id, female),

names\_from = c(variable, year), values\_from = value) %>%

# Transforming factors to factors, all other vars to numeric

mutate\_at(vars(one\_of(factor.vars)), as\_factor) %>%

mutate\_at(vars(-one\_of(factor.vars)), as.numeric)

# Initializing the imputation

init <- mice(period5, maxit = 0, method = "cart", seed = 500) # Set classiffication/regression tree as dflt method

meth <- init$method # Sets the method of imputation for each column

predM <- init$predictorMatrix # Estbalishes the predictor matrix

# Setting id and weight variables in the predictor matrix to zero so that they don't affect the imputation

predM[, c("indiv.id", "perwt\_2017", "perwt\_2019")]= 0

predM[, c("occ2010.ipums\_2017", "occ2010.ipums\_2019", "occ2010\_2017", "occ2010\_2019",

"occ.orig\_2017", "occ.orig\_2019", "ind.orig\_2017", "ind.orig\_2019", "ind1990\_2017", "ind1990\_2019",

"ed.factor\_2019", "ed.factor\_2017", "self.emp.synth\_2019", "self.emp.synth\_2019",

"num.kids.trunc\_2017", "num.kids.trunc\_2019", "dummy.afb.synth\_2017", "dummy.afb.synth\_2019",

"dummy.marstat.synth\_2017", "dummy.marstat.synth\_2019",

"dummy.nkids.synth\_2017", "dummy.nkids.synth\_2019")]= 0

# Running imputation with the specs outlined above

full.imp.p5 <- mice(period5, method = meth, predictorMatrix=predM, m = 5, seed = 500)

# Turning to long

implong.p5 <- mice::complete(full.imp.p5, action = "long", include = TRUE) %>%

# Turnig data back to long format

gather(var, value, -c(.imp, .id, indiv.id, female)) %>%

separate(var, c("var", "year"), "\_20") %>%

mutate(year = case\_when(year == 17 ~ 2017,

year == 19 ~ 2019)) %>%

spread(var, value)

# Merging imputed dataframes

df\_imp.cart <- rbind(implong.p1, implong.p2, implong.p3, implong.p4, implong.p5)

# Saving the imputation to an RDS file

write\_csv(df\_imp.cart, "Output/Clean Data/psid\_imputed\_2019.csv")

rm(df\_imp.cart)

df\_imp.cart <- read\_csv("Output/Clean Data/psid\_imputed\_2019.csv")

# Creating final dataset with auxiliary variables based on imputed variables

# (factors into dummies, log transformations, etc.)

psid\_imp <- df\_imp.cart %>%

arrange(indiv.id, .imp, year) %>%

mutate(

# For race: in the imputation model, race is only ever imputed for one of two years in the same

# "period" (so either 2017 or 1980) as the "wide" values for each year are the same. This means that

# there are some missings in the imputed race data for some respondents in a given year: for one

# respondent in 1980 and for 61 respondents in 2017. For these individuals, we use their assinged

# race in the imputed data in the alternate year in the period (so for the 61 missing cases in 2017,

# we asssign the imputed race in 2015 in that imputed dataset: for the 1 missing case in 1980, we

# assign the imputed race in 1981 in that imputed dataset

race = case\_when(is.na(race) & year == 1980 ~ lead(race),

is.na(race) & year == 1991 ~ lag(race),

is.na(race) & year == 2001 ~ lag(race),

is.na(race) & year == 2011 ~ lag(race),

is.na(race) & year == 2019 ~ lag(race),

TRUE ~ race),

expf = ifelse(expf > expt, expt, expf),

expparttime = expt-expf,

manuf = ifelse(ind.2d %in% c("NonDurable.Manufacturing", "Durable.Manufacturing"), 1, 0),

lnwage = ifelse(wages == 0, 0, log(wages)),

wages.hrly = ifelse(wages == 0 | ann.wrk.hrs == 0, 0, wages/ann.wrk.hrs),

lnhrlywage = ifelse(wages.hrly == 0 | ann.wrk.hrs == 0, 0, log(wages.hrly)),

faminc.minusown = faminc - wages,

agesq = age^2,

agelog = log(age),

expfsq = expf^2,

expparttimesq = expparttime^2,

log.expf = ifelse(expf == 0, 0, log(expf)),

log.expparttime = ifelse(expparttime == 0, 0, log(expparttime)),

ed.factor = ifelse(yrs.ed.fam < 12, "LessthanHS", ifelse(yrs.ed.fam == 12, "HighSchool", ifelse(

yrs.ed.fam > 12 & complete.cases(avdeg) & avdeg == 1, "AdvDeg", ifelse(

yrs.ed.fam > 12 & complete.cases(ba) & ba == 1, "BA", "SomeCollege")))),

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)))),

hrswrk.dummy = case\_when(hrs.wrk.wk < 35 ~ "under35",

hrs.wrk.wk >= 35 & hrs.wrk.wk <= 39 ~ "35to39",

hrs.wrk.wk > 39 & hrs.wrk.wk <= 44 ~ "40to44",

hrs.wrk.wk > 44 & hrs.wrk.wk <= 49 ~ "45to49",

hrs.wrk.wk > 49 ~ "50plus"),

hrswrk.spline1 = ifelse(hrs.wrk.wk < 50, hrs.wrk.wk, 50),

hrswrk.spline2 = ifelse(hrs.wrk.wk >= 50, hrs.wrk.wk-50, 0),

occ.female = ifelse(occ.pct.female >= .70, 1, 0),

occ.male = ifelse(occ.pct.female <= .30, 1, 0),

occ.integrated = ifelse(occ.pct.female < 70 & occ.pct.female > 30, 1, 0),

occ.2d = ifelse(occ2010 <= 430, "Management", ifelse(

occ2010 > 430 & occ2010 <= 950,

"Business.Financial.Operations", ifelse(

occ2010 > 950 & occ2010 <= 1240,

"Computer.Mathematical", ifelse(

occ2010 > 1240 & occ2010 <= 1560,

"Architects.Engineers", ifelse(

occ2010 > 1560 & occ2010 <= 1965,

"Life.Physical.SocialScience", ifelse(

occ2010 > 1965 & occ2010 <= 2060,

"Community.SocialService", ifelse(

# Departing from Census categories to group lawyers

# with physicians, other legal with support

occ2010 %in% c(2100, 2110, 3010, 3060),

"Legal.Physicians.Dentists", ifelse(

occ2010 %in% c(2105, 2145, 2160, 2150, 2140) |

occ2010 >= 2200 & occ2010 <= 2550,

"Education.Training.Library.LegalSupport", ifelse(

occ2010 > 2550 & occ2010 <=2960,

"Arts.Design.Entertainment.Sports.Media", ifelse(

occ2010 > 2960 & occ2010 <= 3540 & occ2010 %!in% c(3010, 3060),

"Health.Practitioners.Technical", ifelse(

occ2010 > 3540 & occ2010 <= 3655,

"Health.Support", ifelse(

occ2010 > 3655 & occ2010 <= 3955,

"Protective.Service", ifelse(

occ2010 > 3955 & occ2010 <= 4160,

"Food.Preparation.Serving", ifelse(

occ2010 > 4160 & occ2010 <= 4250,

"Building.Grounds.Cleaning.Maintenance", ifelse(

occ2010 > 4250 & occ2010 <= 4650,

"Personal.Care.Service", ifelse(

occ2010 > 4650 & occ2010 <= 4965,

"Sales", ifelse(

occ2010 > 4965 & occ2010 <= 5940,

"Office.AdminSupport", ifelse(

occ2010 > 5940 & occ2010 <= 6130,

"Farming.Fishing.Forestry", ifelse(

occ2010 > 6130 & occ2010 <= 6940,

"Construction.Extraction", ifelse(

occ2010 > 6940 & occ2010 <= 7630,

"Installation.Maintenance.Repair", ifelse(

occ2010 > 7630 & occ2010 <= 8965,

"Production", ifelse(

occ2010 > 8965 & occ2010 <= 9750,

"Transportation.Material.Moving", ifelse(

occ2010 > 9750 & occ2010 <= 9830,

"Military", NA)))))))))))))))))))))))) %>%

group\_by(year, .imp) %>%

# Normalizing weight

mutate(perwt\_nonzero = ifelse(perwt == 0, NA, perwt),

perwt\_norm = ifelse(perwt != 0, perwt/mean(perwt\_nonzero, na.rm = T), perwt),

perwt.imm\_nonzero = ifelse(perwt.imm == 0, NA, perwt.imm),

perwt.imm\_norm = ifelse(perwt.imm != 0, perwt.imm/mean(perwt.imm\_nonzero, na.rm = T), perwt.imm)) %>%

dplyr::select(-perwt\_nonzero, -perwt.imm\_nonzero) %>%

mutate(perwt = perwt\_norm,

perwt.imm = perwt.imm\_norm) %>%

ungroup() %>%

dummy\_cols(., select\_columns =

c("ed.factor", "num.kids.trunc", "afb.cat", "ind.2d",

"region", "race", "marstat.synth", "age.youngest.cat",

"hrswrk.dummy", "occ.2d")) %>%

dplyr::select(-"age.youngest.cat\_NA") %>%

rename\_at(vars(starts\_with('ed.factor\_')), funs(sub('ed.factor\_', "", .))) %>%

dplyr::select(-"NA") %>%

rename\_at(vars(starts\_with('ind.2d\_')), funs(sub('ind.2d\_', "", .))) %>%

dplyr::select(-"NA") %>%

rename\_at(vars(starts\_with('region\_')), funs(sub('region\_', "", .))) %>%

dplyr::select(-"NA") %>%

rename\_at(vars(starts\_with('race\_')), funs(sub('race\_', "", .))) %>%

dplyr::select(-"NA") %>%

rename\_at(vars(starts\_with('marstat.synth\_')), funs(sub('marstat.synth\_', "", .))) %>%

rename\_at(vars(starts\_with('num.kids.trunc\_')), funs(sub('num.kids.trunc\_', "numkids.", .))) %>%

dplyr::select(-c("numkids.NA", "afb.cat\_NA", "NA")) %>%

rename\_at(vars(starts\_with('occ.2d\_')), funs(sub('occ.2d\_', "", .))) %>%

dplyr::select(-"NA")

# Saving the final data object as a .csv file

write\_csv(psid\_imp, "Output/Clean Data/psid\_final\_2019.csv")

# Visualiziung missing data

test <- psid\_imp %>%

# Selecting the non-imputed data, individuals meeting sample restrictions

filter(.imp == 0, samp.inc.final == 1) %>%

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

# Selecting variables relevant for the analyses

dplyr::select(year, age, region, race, housework, afb.cat, num.kids.trunc,

marstat.synth, ed.factor, expt, expf, govt.job, union,

ind.2d, occ.orig, occ.pct.female, ann.wrk.hrs, emp.tenure) %>%

# Since all IPUMS-based vars have same levels of missingness, selecting only

# one of them (occ pct female) and labeling it as all census vars

rename(occ.census.vars = occ.pct.female)

vis\_miss(test %>% filter(year == 1981))

vis\_miss(test %>% filter(year == 2019))

# Creating plot of missing data by year

missingbyyr <- psid\_imp %>%

# Selecting the non-imputed data, individuals meeting sample restrictions

filter(.imp == 0, samp.inc.final == 1) %>%

# Selecting variables relevant for the analyses

dplyr::select(year, age, region, race, housework, afb.cat, num.kids.trunc,

marstat.synth, ed.factor, expt, expf, govt.job, union,

ind.2d, occ.orig, occ.pct.female, ann.wrk.hrs, emp.tenure) %>%

# Since all IPUMS-based vars have same levels of missingness, selecting only

# one of them (occ pct female) and labeling it as all census vars

rename(occ.census.vars = occ.pct.female) %>%

group\_by(year) %>%

# Summing the number of missing values by variable by year

summarise\_all(funs(sum(is.na(.)))) %>%

left\_join(., psid\_imp %>% filter(.imp == 0) %>% group\_by(year) %>% summarise(n = n()),

by = "year") %>%

mutate\_all(funs((. / n) \* 100)) %>%

dplyr::select(-n) %>%

mutate(year = c(1980, 1981, 1990, 1991, 1999, 2001, 2009, 2011, 2017, 2019)) %>%

mutate(emp.tenure = ifelse(year == 1980, 0, emp.tenure)) %>%

gather(var, pct.missing, - year) %>%

ggplot() +

geom\_col(aes(x = var, y = pct.missing, fill = "Percent Missing")) +

labs(title = "Missing Data by Year (All Sample Restrictions)",

y = "Percent Missing",

x = "Variables") +

theme\_bw() +

facet\_wrap(~year, scales = "free\_x") +

theme(legend.position = "none",

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

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

ggsave("Output/Figures/misssingbyyr.pdf", plot = missingbyyr, height = 10, width = 12, units = "in")

toc()