#Diego Osorio, Samuel Malkún y Daniel Franco.

#Set de preguntas

######Punto 2

#Log o no, si utilizamos log borrar NA o qué hacer?

#La variable inglabo suena bien o debería ser ingreso total, podemos usar ambas? ( hay nuevas variables y\_income)

#¿Cómo se hace la discusión del model´s in sample fit? ¿Es el R^2, F, MSE, F\_1?

#Es un plot de los coeficientes? o qué plot es el que se está pidiendo?

#Ya tenemos el peak age, pero cómo se hace el intervalo de confianza? ¿También quieren los intervalos de los betas?¿Asumimos dist normal?

####Punto 3

#acá es obligatorio logaritmo? o se compara con y sin?

#Misma duda que el 2 con el plot y los intervalos

# 1. importar librarias y set environment/wd

# llamar la librería pacman: contiene la función p\_load()

rm(list=ls())

#Establecer directorio de trabajo

setwd("~/Desktop/Big Data/Repositorios/BD-ML---PS1")

set.seed(1000)

require(pacman)

p\_load(tidyverse, # contiene las librerías ggplot, dplyr...

rvest, data.table, dplyr, skimr, # summary data

caret, rio, vtable, stargazer, ggplot2, boot, MLmetrics, lfe) # web-scraping

#Instalar paquetes

install.packages("data,table")

#Cargar librerías

library(data.table)

# 2. web-scrapping: descargar URL con databases y construir bases

####Descargar los chunks####

# cargar data-set

# creamos una lista que contiene todas las urls de cada datachunk

lista\_urls <- paste0("https://ignaciomsarmiento.github.io/GEIH2018\_sample/pages/geih\_page\_",1:10,".html")

# crear dataframe vacio - estructura de datos

df <- data.frame()

# iterar sobre la lista de urls

for (url in lista\_urls) {

# print url actual que contiene el data chunk

print(url)

# read\_html(): leer url del data-chunk

# html\_table(): crear tabla html

temporal <- read\_html(url)%>%html\_table()

# as.data.frame() crea un data-frame temporal

df\_temportal <- as.data.frame(temporal[[1]])

# rbind(df1,df2): unir el df general con el df temporal

df <- rbind(df, df\_temportal)

}

# 3. Data mining and data cleaning

# %>% - (df.fun1.fun2 in Python)

# df%>%subset: overwrite sobre df on condition ocu == 1 & age >=18

df2 <- df%>%subset(ocu == 1 & age >=18)

#Url de la complementaria 1

browseURL("https://lectures-r.gitlab.io/big-data-202202/week-01/")

# guardar/escribirlas bases de datos en csv en el wd establecido

write.csv(df2, "datafiltrada.csv")

write.csv(df, "dataoriginal.csv")

# read\_csv(file): cargar database

df <- read.csv("dataoriginal.csv")

df2 <- read.csv("datafiltrada.csv")

# 4. Estadísticas descriptivas: análisis estadístico y descripción de variables

## summary db

names(df2)[names(df2) == ''] <- 'Índice'

# Se guarda como tibble

db <- as\_tibble(df2)

# skim

skim(db)%>%head()

head(db$sex, 100)

#Tabla de estadísticas descriptivas con stargazer

stargazer(df2[c("ingtot", "age")], type = "html", title = "Estadísticas Descriptivas", out = "estdec.html")

#Gráficos#

# boxplot estrato vs ingreso total g1

ggplot(df2, aes(x = as.factor(estrato1) , y = log(ingtot) , fill = as.factor(sex))) +

geom\_boxplot()+

scale\_fill\_hue(l=60, c=80)+

ggtitle("Ln del ingreso total según estrato social y sexo")+

xlab("Estrato o ICV")+

ylab("Ln del ingreso total")+

theme(plot.title = element\_text(hjust = 0.5))+

labs(fill = "Sexo")+

theme\_classic()+

theme(text = element\_text(size = 16), plot.title = element\_text(hjust = 0.5, size = 20), legend.text = element\_text(size = 14), legend.title = element\_text(size = 16))+

scale\_fill\_manual(values = c("0" ="red" , "1"="blue"), label = c("0" ="Mujer" , "1"="Hombre"))

# scatter sin puntos de edad vs ingreso total g2

ggplot(df2, aes(x = age, y = log(ingtot)))+

geom\_smooth(method = "loess", level = 0.95, aes(weight = fex\_c))+

ggtitle("Ingreso total según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

# scatter edad vs impa g3

ggplot(df2, aes(x = age, y = log(impa)))+

geom\_smooth(method = "loess", level = 0.95, aes(weight = fex\_c))+

ggtitle("Ingreso monetario de primera actividad según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso monetario de primera actividad")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

# scatter con puntos de edad vs ingreso total b4

ggplot(df2, aes(x = age, y = log(ingtot)))+

geom\_point()+

geom\_smooth(method = "loess", level = 0.95, aes(weight = fex\_c))+

ggtitle("Ingreso total según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Scatter del y\_laboral vs edad

ggplot(df2, aes(x = age, y = log(y\_ingLab\_m)))+

geom\_point()+

geom\_smooth(method = "loess", level = 0.95, aes(weight = fex\_c))+

ggtitle("Perfil de ingreso laboral y edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso laboral mensual")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

# 5. Modelos de Regresion Lineal

#relación ingreso y edad

# mutate(new\_col=f(var)): crea nueva columna a partir de existentes

# mutate(x=var):permite crear nuevas variables a partir de otras variables para construir funciones f(y), f(x)

df2 <- df2%>%mutate(age2=age^2)

df2 <- df2%>%mutate(inglabo=impa+isa)

### Bootstraping ###

# boot(database, eta\_fun, R=N): permite obtener los estimadores t(beta) del modelo

eta.fn\_1<-function(data,index){

coef(lm(y\_ingLab\_m~age+age2, data = df2, weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

boot1 <- boot(df2, eta.fn\_1, R = 1000)

output\_boot1 <- t(rbind(boot1$t0, apply(boot1$t, 2, function(x) sd(x))))

eta.fn\_11<-function(data,index){

f<-lm(y\_ingLab\_m~age+age2, data = df2, weights = fex\_c, subset = index)

coefs\_11 <- f$coefficients

b2<-coefs\_11[2]

b3<-coefs\_11[3]

age\_peak\_nolog<- -b2/(2\*b3)

return(age\_peak\_nolog)

}

# boot(data, eta\_func, R=N)

boot11 <- boot(df2, eta.fn\_11, R = 1000)

output\_boot11 <- t(rbind(boot11$t0, apply(boot11$t, 2, function(x) sd(x))))

alpha <- 0.05

age\_peak\_min\_nolog <- output\_boot11[1,1]-qnorm(1-(alpha/2))\*output\_boot11[1,2]

age\_peak\_max\_nolog <- output\_boot11[1,1]+qnorm(1-(alpha/2))\*output\_boot11[1,2]

df2 <- df2%>%mutate(inglabo\_hat\_nolog = boot1$t0[1]+boot1$t0[2]\*age+boot1$t0[3]\*age2)

ggplot(df2, aes(x = age, y = inglabo\_hat\_nolog/1000))+

geom\_line()+

ggtitle("Perfil ingreso laboral y edad")+

xlab("Edad")+

ylab("Ingreso laboral estimado en miles")+

theme\_classic()+

ylim(200,2500)+

geom\_vline(xintercept = age\_peak\_min\_nolog, linetype="dotted", color = "red", size=1)+

geom\_vline(xintercept = age\_peak\_max\_nolog, linetype="dotted", color = "red", size=1)+

geom\_vline(xintercept = output\_boot11[1,1], linetype="dotted", color = "blue", size=1)+

geom\_text(aes(x=48, label="\nEdad pico=48.5 ", y=1000), colour="blue", angle=90)+

geom\_text(aes(x=45.8, label="\nRango minimo=48.5", y=1000), colour="red", angle=90)+

geom\_text(aes(x=51.2, label="\nRango máximo=51.2", y=1000), colour="red", angle=90)+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

eta.fn\_2<-function(data,index){

coef(lm(log(y\_ingLab\_m)~age+age2, data = df2, weights = fex\_c, subset = index))

}

boot2 <- boot(df2, eta.fn\_2, R = 1000)

output\_boot2 <- t(rbind(boot2$t0, apply(boot2$t, 2, function(x) sd(x))))

eta.fn\_22<-function(data,index){

f<-lm(log(y\_ingLab\_m)~age+age2, data = df2, weights = fex\_c, subset = index)

coefs\_22 <- f$coefficients

b2<-coefs\_22[2]

b3<-coefs\_22[3]

age\_peak\_log<- -b2/(2\*b3)

return(age\_peak\_log)

}

# boot(data, eta\_func, R=N)

boot22 <- boot(df2, eta.fn\_22, R = 1000)

output\_boot22 <- t(rbind(boot22$t0, apply(boot22$t, 2, function(x) sd(x))))

alpha <- 0.05

age\_peak\_min\_log <- output\_boot22[1,1]-qnorm(1-(alpha/2))\*output\_boot22[1,2]

age\_peak\_max\_log <- output\_boot22[1,1]+qnorm(1-(alpha/2))\*output\_boot22[1,2]

df2 <- df2%>%mutate(inglabo\_hat\_log = boot2$t0[1]+boot2$t0[2]\*age+boot2$t0[3]\*age2)

# Gráfico

ggplot(df2, aes(x = age, y = inglabo\_hat\_log))+

geom\_line()+

ggtitle("Perfil ingreso laboral y edad")+

xlab("Edad")+

ylab("Ingreso laboral estimado en logaritmo")+

theme\_classic()+

geom\_vline(xintercept = age\_peak\_min\_log, linetype="dotted", color = "red", size=1)+

geom\_vline(xintercept = age\_peak\_max\_log, linetype="dotted", color = "red", size=1)+

geom\_vline(xintercept = output\_boot22[1,1], linetype="dotted", color = "blue", size=1)+

geom\_text(aes(x=43.1, label="\nEdad pico=43.1 ", y=10), colour="blue", angle=90)+

geom\_text(aes(x=42.3, label="\nRango minimo=42.3", y=10), colour="red", angle=90)+

geom\_text(aes(x=43.9, label="\nRango máximo=43.9", y=10), colour="red", angle=90)+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Sacar la tabla

stargazer(output\_boot1, type = "html", title = "Ingreso laboral", out = "mod1.html")

stargazer(output\_boot2, type = "html", title = "Logaritmo del Ingreso laboral", out = "mod2.html")

#Age peak: igualar la función derivada de y = 0

age\_peak1 <- -(boot1$t0[2]/(2\*boot1$t0[3]))

age\_peaklog <- -(boot2$t0[2]/(2\*boot2$t0[3]))

eta.fn\_3<-function(data,index){

coef(lm(inglabo~age+age2, data = df2, weights = fex\_c, subset = index))

}

boot3 <- boot(df2, eta.fn\_3, R = 1000)

eta.fn\_4<-function(data,index){

coef(lm(log(inglabo+1)~age+age2, data = df2, weights = fex\_c, subset = index))

}

boot4 <- boot(df2, eta.fn\_4, R = 1000)

# mutate(x=var):permite crear nuevas variables a partir de otras variables para construir funciones f(y), f(x)

# inglabo\_hat(fun\_predic), inglabo\_hat\_nolog(fun\_predic)

df2 <- df2%>%mutate(inglabo\_hat = exp(boot2$t0[1]+boot2$t0[2]\*age+boot2$t0[3]\*age2))

df2 <- df2%>%mutate(inglabo\_hat\_nolog = boot1$t0[1]+boot1$t0[2]\*age+boot1$t0[3]\*age2)

# gráfico solo con valores predichos g5

ggplot(df2, aes(x = age, y = log(inglabo\_hat)))+

geom\_line()+

ggtitle("Perfil ingreso laboral y edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso laboral estimado")+

theme\_classic()+

geom\_vline(xintercept = 43.1, linetype="dotted", color = "red", size=1.5)+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

ggplot(df2, aes(x = age, y = inglabo\_hat\_nolog/1000))+

geom\_line()+

ggtitle("Perfil ingreso laboral y edad")+

xlab("Edad")+

ylab("Ingreso laboral estimado en miles")+

theme\_classic()+

ylim(200,2500)+

geom\_vline(xintercept = 48.5, linetype="dotted", color = "red", size=1)+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Gráfico con ambos g6

ggplot(data = df2)+

geom\_point(mapping = aes(x = age, y = log(inglabo)), color = "blue")+

geom\_point(mapping = aes(x = age, y = log(inglabo\_hat)), color = "red")+

geom\_point(mapping = aes(x = age, y = log(inglabo\_hat\_nolog)), color = "orange")+

ggtitle("Ingreso total según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Gráfico sin logaritmos g7

ggplot(data = df2)+

geom\_point(mapping = aes(x = age, y = (inglabo)), color = "blue")+

geom\_point(mapping = aes(x = age, y = (inglabo\_hat)), color = "red")+

geom\_point(mapping = aes(x = age, y = (inglabo\_hat\_nolog)), color = "orange")+

ggtitle("Ingreso total según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Age peak: igualar la función derivada de y = 0

age\_peak <- -(boot2$t0[2]/(2\*boot2$t0[3]))

age\_peak\_nolog <- -(boot3$t0[2]/(2\*boot3$t0[3]))

#Obtener errores estándar de los estimadores

output\_tab <- t(rbind(boot3$t0, apply(boot3$t, 2, function(x) sd(x))))

# Intervalos de confianza

alpha <- 0.05

age\_peak\_min\_nolog <- -((boot3$t0[2]- qnorm(alpha/2)\*output\_tab[2,2])/(2\*(boot3$t0[3]- qnorm(alpha/2)\*output\_tab[3,2])))

age\_peak\_max\_nolog <- -((boot3$t0[2]+ qnorm(alpha/2)\*output\_tab[2,2])/(2\*(boot3$t0[3]+ qnorm(alpha/2)\*output\_tab[3,2])))

####Punto 3####

#Crear la variable female

# mutate(var=dep)

df2 <- df2%>%mutate(female = 1-sex)

p3m1 <- lm(data = df2, log(ingtot+1) ~ female, weights = fex\_c)

summary(p3m1)

p3m2 <- lm(data = df2%>%subset(ingtot > 0), log(ingtot) ~ female, weights = fex\_c)

summary(p3m2)

#Age earnings by gender

p3m3 <- lm(data = df2, log(ingtot+1) ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m3)

p3m4 <- lm(data = df2%>%subset(ingtot > 0), log(ingtot) ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m4)

p3m5 <- lm(data = df2, ingtot ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m5)

#Cambiando ingtot por inglabo

p3m6 <- lm(data = df2, log(inglabo+1) ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m6)

# modelo regresion lineal

# f(y) = log(inglabo), inglabo=impa+isa

# f(x) = female\*age+female\*age2

# df2%>%subset(inglabo > 0) overwrite df2 on condition:inglabo > 0

# factor de expansión: weights = fex\_c

p3m7 <- lm(data = df2%>%subset(inglabo > 0), log(inglabo) ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m7)

p3m8 <- lm(data = df2, inglabo ~ female\*age+female\*age2, weights = fex\_c)

summary(p3m8)

### Bootstraping ###

# bootstrap para p3m7

eta.fn\_p3m7<-function(data,index){

coef(lm(log(inglabo) ~ female\*age+female\*age2, data = df2%>%subset(inglabo>0), weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp3m7 <- boot(df2, eta.fn\_p3m7, R = 1000)

eta.fn\_p3m1<-function(data,index){

coef(lm(log(ingtot+1) ~ female, data = df2, weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp3m1 <- boot(df2, eta.fn\_p3m1, R = 1000)

eta.fn\_p3m2<-function(data,index){

coef(lm(log(ingtot) ~ female, data = df2%>%subset(ingtot > 0), weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp3m2 <- boot(df2, eta.fn\_p3m2, R = 1000)

eta.fn\_p3m3<-function(data,index){

coef(lm(log(ingtot+1) ~ female\*age+female\*age2, data = df2, weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp3m3 <- boot(df2, eta.fn\_p3m3, R = 1000)

eta.fn\_p4m4<-function(data,index){

coef(lm(log(ingtot) ~ female\*age+female\*age2, data = df2%>%subset(ingtot > 0), weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp4m4 <- boot(df2, eta.fn\_p4m4, R = 1000)

eta.fn\_p5m5<-function(data,index){

coef(lm(ingtot ~ female\*age+female\*age2, data = df2, weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootp5m5 <- boot(df2, eta.fn\_p5m5, R = 1000)

## graficas: scatter plots

# f(y) = log(inglabo), inglabo=impa+isa

# f(x) = female\*age+female\*age2

# df2%>%subset(inglabo > 0) overwrite df2 on condition:inglabo > 0

# crear variables para predictores

# mutate(x=var):permite crear nuevas variables a partir de otras variables para construir funciones f(y), f(x)

# inglabo\_hat(fun\_predic), inglabo\_hat\_nolog(fun\_predic)

# inglaboFem\_hat\_m: funcion prediccion de inglaboFem

df2 <- df2%>%mutate(inglaboFem\_hat\_m = exp(bootp3m7$t0[1]+bootp3m7$t0[2]\*female+bootp3m7$t0[3]\*age+bootp3m7$t0[4]\*age2+bootp3m7$t0[5]\*female\*age+bootp3m7$t0[6]\*female\*age2))

# gráfico solo con valores predichos g5

ggplot(df2, aes(x = age, y = log(inglaboFem\_hat\_m), group=as.factor(female), color=as.factor(female)) ) +

geom\_point()+

ggtitle("Ingreso total según la edad, por sexo")+

xlab("Edad")+

labs(colour = "Sexo")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )+

scale\_colour\_manual(values=c('0'='blue', '1'='red'), label= c('0'='Hombre', '1'='Mujer') )

# gráfico solo con valores originales g6

ggplot(df2%>%subset(inglabo>0), aes(x = age, y = log(inglabo), group=as.factor(female), color=as.factor(female)) ) +

geom\_point()+

ggtitle("Ingreso total según la edad, por sexo")+

xlab("Edad")+

labs(colour = "Sexo")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )+

scale\_colour\_manual(values=c('0'='blue', '1'='red'), label= c('0'='Hombre', '1'='Mujer') )

#Gráfico con ambos g6

ggplot(data = df2%>%subset(inglabo>0))+

geom\_point(mapping = aes(x = age, y = log(inglabo)), color = "blue")+

geom\_point(mapping = aes(x = age, y = log(inglaboFem\_hat\_m)), color = "red")+

#geom\_point(mapping = aes(x = age, y = log(inglabo\_hat\_nolog)), color = "orange")+

ggtitle("Ingreso total según la edad")+

xlab("Edad")+

ylab("Logaritmo del ingreso total")+

theme\_classic()+

theme(plot.title = element\_text(hjust = 0.5, size = 20), axis.title.x = element\_text(hjust = 0.5, size = 16), axis.title.y = element\_text(hjust = 0.5, size = 16), axis.text = element\_text(size = 14) )

#Age peak: igualar la función derivada de y = 0

# caso hombres

age\_peak\_hom <- -(bootp3m7$t0[3])/(2\*bootp3m7$t0[4])

# caso mujeres

age\_peak\_fem <- -(bootp3m7$t0[3]+bootp3m7$t0[5])/(2\*(bootp3m7$t0[4]+bootp3m7$t0[6]))

### FALTA DETERMINAR LA FUNCION PARA CALCULAR LOS INTERVALOS DE CONFIANZA

#Sacar los intervalos para el peak age con sus IC

#Obtener errores estándar de los estimadores\*\*\*\*

output\_tab <- t(rbind(boot3$t0, apply(boot3$t, 2, function(x) sd(x))))

# Intervalos de confianza

alpha <- 0.05

age\_peak\_min\_nolog <- -((boot3$t0[2]- qnorm(alpha/2)\*output\_tab[2,2])/(2\*(boot3$t0[3]- qnorm(alpha/2)\*output\_tab[3,2])))

age\_peak\_max\_nolog <- -((boot3$t0[2]+ qnorm(alpha/2)\*output\_tab[2,2])/(2\*(boot3$t0[3]+ qnorm(alpha/2)\*output\_tab[3,2])))

#Punto 3 c

names(df2)

# contruir nueva base para gap con variables de control

dfbase<- df2[c("age", "age2", "female", "clase", "p6210", "relab","p6210s1", "college", "cotPension", "cuentaPropia", "estrato1", "fex\_c", "formal", "fweight", "hoursWorkUsual", "inglabo", "ingtot", "impa", "isa", "maxEducLevel", "mes", "microEmpresa", "oficio", "p6050","p6426", "y\_ingLab\_m", "y\_ingLab\_m\_ha", "y\_salarySec\_m", "y\_salary\_m\_hu", "y\_total\_m", "y\_total\_m\_ha")]

write.csv(dfbase, "datagap.csv")

dfgap <- read.csv("datagap.csv")

# modelo lineal - variables de control: maxEducLevel

modelo\_gap1 <- lm(data = dfgap, y\_total\_m ~ age\*female+age2\*female+factor(maxEducLevel) , weights = fex\_c)

summary(modelo\_gap1)

# modelo lineal - variables de control: maxEducLevel, factor(estrato1)

modelo\_gap2 <- lm(data = dfgap, y\_total\_m ~ age\*female+age2\*female+factor(maxEducLevel)+ factor(estrato1) , weights = fex\_c)

summary(modelo\_gap1)

# modelo lineal - variables de control: maxEducLevel, factor(estrato1)

modelo\_gap3 <- lm(data = dfgap, y\_total\_m ~ age\*female+age2\*female+factor(maxEducLevel)+ factor(estrato1) , weights = fex\_c)

summary(modelo\_gap1)

# modelo lineal - variables de control: nivel educativo, estrato1, relab, oficio1

# basado en ncbi:literatura

dfgap2 <- dfgap%>%subset(y\_ingLab\_m>0)

modelo\_gap4 <- lm(data = dfgap2, log(y\_ingLab\_m) ~ age\*female+age2\*female+factor(maxEducLevel)+ factor(estrato1) + factor(relab), weights = fex\_c)

summary(modelo\_gap4)

# modelo fwl (matriz de proyección y annihilation)

modelo\_gap4\_fwl <- felm(log(y\_ingLab\_m) ~ age\*female+age2\*female| factor(maxEducLevel)+ factor(estrato1)+ factor(relab), data =dfgap2, weights = dfgap2$fex\_c)

summary(modelo\_gap4\_fwl)

# bootstping para modelo con fwl (matriz de proyección y annihilation)

eta.fn\_gap4<-function(data,index){

coef(lm(log(y\_ingLab\_m) ~ age\*female+age2\*female+factor(maxEducLevel)+ factor(estrato1) + factor(relab), data = dfgap2, weights = fex\_c, subset = index))

}

# boot(data, eta\_func, R=N)

bootgap4 <- boot(dfgap2, eta.fn\_gap4, R = 1000)

eta.fn\_gap5<-function(data,index){

coef(lm(log(y\_ingLab\_m) ~ age\*female+age2\*female, data = dfgap2, weights = fex\_c, subset = index))

}

bootgap5 <- boot(dfgap2, eta.fn\_gap5, R = 1000)

bootgap5

eta.fn\_gap6<-function(data,index){

coef(felm(log(y\_ingLab\_m) ~ age\*female+age2\*female| factor(maxEducLevel)+ factor(estrato1)+ factor(relab), data = dfgap2, weights = dfgap2$fex\_c, subset = index))

}

bootgap6 <- boot(dfgap2, eta.fn\_gap6, R = 1000)

bootgap6

## Punto 4: Prediction and Performance Evaluation

# prediction, overfitting and cross-val

# split database into test-set and test-set

p\_load(tidyverse, fabricatr, stargazer)

# set/crear seed para reproducibilidad

set.seed(101010)

# crear dataframe para fit/train el modelo

df\_ml <- df2%>%mutate(holdout= as.logical(1:nrow(df2)%in%sample(nrow(df2), nrow(df2)\*.3)) )

test\_set <- df\_ml[df\_ml$holdout==T,]

train\_set <- df\_ml[df\_ml$holdout==F,]

# contruir el modelo dummy/naive especificacion

spec1 <- lm()