**Replicability Report**

Hello Samuel,

I followed your codes and instructions line by line. I had error messages in some instances and tried to correct them but I eventually ended up with a faulty dataset, could be my mistake trying to fix the error messages. The text has been colour-coded: Code = black, Your comments = green, My comments = blue, Identified error = red.

Script

census <- read.csv("G:/RStudio Files/Census\_Data.csv") # sets working directory

If you created a project directory with the dataset in the folder, read.csv (“”) function would have imported the file without stating the file path. Eg. census <- read.csv("Census 2016.csv")

colnames(census) <- c("PPSOR", "HDGREE", "AGEIMM", "IMMSTAT", "POB", "PR", "SEX", "TOTINC")

Other variable names became “NA”, I continued. (I later realised that this variable selection severely distorted my dataset especially the DV – TotInc making me run into several errors and unexpected regression result)

# Load necessary libraries

library(dplyr)

library(ggplot2)

# Remove rows with TOTINC of 88,888,888 or 99,999,999, as they count as invalid data here.

census\_clean <- census %>%

filter(!TOTINC %in% c(88888888, 99999999))

I got an error message

**in `filter()`:** Can't transform a data frame with `NA` or `""` names. This was referring to column names that are now NA, but you specified TOTINC, I don’t know why it reads other columns.

So I check the column names.

Names (census)

I asked ChatGPT to help me with code to remove columns with NA or empty names using dplyr

census\_clean <- census %>%

select(-which(is.na(names(census)) | names(census) == ""))

# Repeat code and it worked

census\_clean <- census\_clean %>%

filter(!TOTINC %in% c(88888888, 99999999))

# Recoding POB into a binary variable (1 for outside Canada, 0 for born in Canada)

census\_clean <- census\_clean %>%

mutate(POB\_Binary = ifelse(POB ==1, 0, 1))

# Recoding IMMSTAT into Binary variable (1 for immigrant, 0 for non-immigrant)

census\_clean <- census\_clean %>%

mutate(ImmigrationStat = ifelse(IMMSTAT ==1, 0, 1))

# I would like to see the number of non-immigrants born outside Canada, for nothing other than curiosity.

non\_immigrants\_outside\_canada <- census\_clean %>%

filter(POB\_Binary == 1 & IMMSTAT == 1) %>%

count()

print(non\_immigrants\_outside\_canada)

You didn’t say what was the result. I got n =2754.

# We can now get into the recoding of income - it is in the census as a continuous variable, but I would like it to be broken down into categories with increments of $20,000.

census\_clean <- census\_clean %>%

mutate(TOTINC\_category = case\_when(

TOTINC <= 19999 ~ 0,

TOTINC >= 20000 & TOTINC <= 49999 ~ 1,

TOTINC >= 40000 & TOTINC <= 59999 ~ 2,

TOTINC >= 60000 & TOTINC <= 79999 ~ 3,

TOTINC >= 80000 & TOTINC <= 99999 ~ 4,

TOTINC >= 1000000 & TOTINC <= 119999 ~ 5,

TOTINC >= 120000 & TOTINC <= 139999 ~ 6,

TOTINC >= 140000 ~ 7,

TRUE ~ NA\_real\_

))

Code failed here with an error message **in `filter()`:** ! Can't transform a data frame with `NA` or `""` names. Run `rlang::last\_trace()` to see where the error occurred. I tried to run the next part together but it didn’t work.

I discovered that the upper and lower cases in the variable names in this code differ from what we have in the imported csv. File (see TotInc ≠ TOTINC, Sex ≠ SEX)

# I would also, for the sake of the logistic regression I will run in a moment, create a binary income variable, representing top earners versus non-top earners. This will be an arbitrary number for the sake of this exercise, though if my memory serves me, it is around the $80,000 mark.

census\_clean <- census\_clean %>%

mutate(topearner = ifelse(TOTINC >= 80000, 1, 0))

# I would also like to recode HDGREE into a slightly nicer variable,where 1=no degree, 2=degree below bachelors, 3=bachelors degree or above, 4=masters, 5=phd, and 6=medicine degree, while 99 indicates no response or not applicable.

census\_clean <- census\_clean %>%

mutate(DegreeEarned = case\_when(

HDGREE == 1 ~ 1,

HDGREE %in% 2:8 ~ 2,

HDGREE %in% 9:10 ~ 3,

HDGREE == 12 ~ 4,

HDGREE == 13 ~ 5,

HDGREE == 11 ~ 6,

HDGREE %in% c(88, 99) ~ 99,

TRUE ~ NA\_real\_

))

# Summary table of immigrants vs non-immigrants making above $80,000 to visualize.

summary\_table <- census\_clean %>%

filter(topearner >= 0) %>%

group\_by(POB\_Binary, ImmigrationStat) %>%

summarise(count = n()) %>%

ungroup()

print(summary\_table)

#With everything recoded, we can get to our binary logistic regression to look at the dependent variable (top earner or no) and if and how it is affected by the individuals place of birth, immigration status, gender, degree, province of residence, and age at immigration.

logit\_model <- glm(topearner ~ POB\_Binary + ImmigrationStat + SEX + DegreeEarned + PR + AGEIMM,

data = census\_clean,

family = "binomial")

#summary of the regression

summary(logit\_model)

nrow(census\_clean)