```r
################################################################################
##Packages
################################################################################

library(foreign)
library(survey)
library(dplyr)
library(tidyr)
library(survey)
library(gridExtra)
library(ggplot2)
library(Hmisc)
################################################################################
## Import data from NHANES 2017 - 2018
################################################################################

milk <- read.xport("/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/DBQ_J.xpt")
choles <- read.xport("/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/TRIGLY_J.xpt")
demo <- read.xport("/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/DEMO_J.xpt")

# save as an R data frame
saveRDS(milk, file="/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/DEMO_I.rds")
saveRDS(choles, file="/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/TRIGLY_J.rds")
saveRDS(demo, file="/Users/huongtran/OU /Course Work/SES 3/STA5229/Final
Project/dataset/DEMO_J.rds")

################################################################################
## Merging data: keep all records in DEMO, even though SEQN does not match
################################################################################
df <- left_join(select(demo, "SEQN", starts_with("SDMV"), "WTMEC2YR"),
                select(milk, "SEQN", starts_with("DBQ223")), by="SEQN")
%>%
  left_join(., select(choles, "SEQN", "LBDLDL") , by="SEQN")

#descrbie milk data:
describe(select(df, "DBQ223A", "DBQ223B", "DBQ223C", "DBQ223D", "DBQ223E",
"DBQ223U"))


for (i in 1:nrow(df) ){
  if (is.na(df$DBQ223A[i]) == T & is.na(df$DBQ223B[i]) == T &
is.na(df$DBQ223C[i]) == T &
      is.na(df$DBQ223D[i]) == T & is.na(df$DBQ223E[i]) == T &
is.na(df$DBQ223U[i]) == T)  {
    df$Category[i] = NA}
  else {
    t <- sum(df$DBQ223A[i], df$DBQ223B[i], df$DBQ223C[i], df$DBQ223D[i],
df$DBQ223E[i], df$DBQ223U[i], na.rm = T)
    if (t== 10) {
      df$Category[i] <- "whole milk"
    } else if ( t == 99) {
      df$Category[i] <- "don't know"
```

```r
    } else if (t == 11) {
      df$Category[i] <- "2% fat"
    } else if (t == 12) {
      df$Category[i] <- "1% fat"
    } else if (t == 13) {
      df$Category[i] <- "fat free"
    } else if (t == 14) {
      df$Category[i] <- "soy milk"
    } else if (t == 30) {
      df$Category[i] <- "another"
    } else {
      df$Category[i] <- "2 types"
    }
  }
}


df <- select(df, - starts_with("DBQ223"))


############################################################################
# Create survey design object:
############################################################################
sv <- svydesign(id=~SDMVPSU,  strata =~SDMVSTRA,
                data=df, weights=~WTMEC2YR,nest=TRUE)

deff(df$LBDLDL, cluster = df$SDMVPSU)


############################################################################
#Ananlysis
############################################################################
#estimate polulation cumulative distribution:
plot(svycdf(~LBDLDL, sv),
     main= "Population of LDL Cholesterol level", col = "blue")
svyplot(~LBDLDL, sv, main = "Scatterplot for sampling weights", col =
"blue")


############################################################################
sub.sv <- subset(sv, is.na(df$Category) == F &
                   df$Category != "2 types" &
                   df$Category != "don't know")

svyboxplot(LBDLDL ~ Category, design = sub.sv,
           main = "Boxplot of LDL Cholesterol level/each types of milk")

mean <- svyby(~LBDLDL,~Category, design =  sub.sv,
              svymean, na.rm = T)
result <- select(cbind(mean, confint(mean)), -"Category")

############################################################################
# Regression
############################################################################
model <- svyglm(LBDLDL ~ Category, design = sub.sv )
s <- summary(model)$coefficients
print(s, digits = 2)
```

```
###########################################################################
#test whole milk/fat free milk
###########################################################################
sub.sv2 <- subset(sv, df$Category == "whole milk" |
                      df$Category == "fat free")
svyttest(LBDLDL~Category, sub.sv2)
```