final

Library

```
library(tidyverse)
library(haven)
library(NHANES)
library(dplyr)
library(purrr)
library(lubridate)
library(glmnet)
library(vip)
library(car)
library(caret)
library(randomForest)
library(doParallel)
library(mice)
library(naniar)
```

Dataset

2020

Primary Question

```
getwd()
setwd("./data/2020")

ALQ <- read_xpt("P_ALQ.XPT")
```

```r
BIOPRO <- read_xpt("P_BIOPRO.XPT")
BMX <- read_xpt("P_BMX.XPT")
BPXO <- read_xpt("P_BPXO.XPT")
DEMO <- read_xpt("P_DEMO.XPT")
#FOLATE <- read_xpt("P_FOLATE.XPT")
PAQ <- read_xpt("P_PAQ.XPT")
SLQ <- read_xpt("P_SLQ.XPT")
TCHOL <- read_xpt("P_TCHOL.XPT")
DPQ_raw <- read_xpt("P_DPQ.XPT")


DPQ <- DPQ_raw %>%
    filter(complete.cases(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_score = rowSums(select(., starts_with("DPQ"))[-1])) |> #%>%
    mutate(depression_category = case_when(
        depression_score <= 4 ~ "Minimal",
        depression_score <= 9 ~ "Mild",
        depression_score <= 14 ~ "Moderate",
        depression_score <= 19 ~ "Moderately severe",
        depression_score >= 20 ~ "Severe"
    ))

BPXO <- BPXO %>%
  mutate(
    MAP1 = (1 / 3) * BPXOSY1 + (2 / 3) * BPXODI1,
    MAP2 = (1 / 3) * BPXOSY2 + (2 / 3) * BPXODI2,
    MAP3 = (1 / 3) * BPXOSY3 + (2 / 3) * BPXODI3,
    avg_MAP = rowMeans(cbind(MAP1, MAP2, MAP3), na.rm = TRUE)
  )

datasets <- list(ALQ, BIOPRO, BMX, BPXO, DEMO,  PAQ, SLQ, TCHOL, DPQ)
test <- reduce(datasets, inner_join, by = "SEQN")
temp <- select(test, SEQN , BMXBMI, avg_MAP, ALQ130, RIAGENDR, RIDRETH3, RIDAGEYR, INDFMPIR,

dat_raw_2020 <- left_join(SLQ, temp)

new_colnames <-  c(
  "ID",                          # SEQN; Respondent sequence number
  "sleep_time_weekdays",       # SLQ300;Usual_sleep_time_weekdays
  "wake_time_weekdays",        # SLQ310;Usual_wake_time_weekdays
  "sleep_hours_weekdays",            # SLD012;Sleep_hours_weekdays
  "sleep_time_weekends",       # SLQ320;Usual_sleep_time_weekends
```

```
  "wake_time_weekends",        # SLQ330;Usual_wake_time_weekends
  "sleep_hours_weekends",           # SLD013;Sleep_hours_weekends
  "frq_snore",                 # SLQ030;How_often_snore
  "frq_snort_or_stop_breathing",# SLQ040;How_often_snort_or_stop_breathing
  "sleep_trouble",  # SLQ050;Ever_told_doctor_sleep_trouble
  "overly_sleepy",          # SLQ120;Feel_overly_sleepy_day
  "BMI",                       # BMXBMI; Body Mass Index
  "avg_MAP",                   # avg_MAP; Average Mean Arterial Pressure
  "alcohol",                   # ALQ130; How often drink alcohol
  "gender",                    # RIAGENDR; Gender
  "race_ethnicity",        # RIDRETH1; Race/Hispanic origin
  "age",                       # RIDAGEYR; Age at screening
  "income",  # INDFMPIR; Family income to poverty ratio
  "sedentary_activity",        # PAD680; Minutes sedentary activity
  "total_cholesterol",         # LBDTCSI; Total cholesterol mmol/L
  "depression_score",          # depression_score; Depression score
  "depression_category"        # depression_category; Depression category
)
colnames(dat_raw_2020) <- new_colnames
#remove all lines with NA
dat_raw_2020[dat_raw_2020 == ""] <- NA
dat_2020 <- dat_raw_2020 |>
  drop_na() |>
   filter(alcohol < 16 & frq_snore != 7 & frq_snore != 9 & frq_snort_or_stop_breathing != 7 &
  mutate(across(c(sleep_time_weekdays,wake_time_weekdays,sleep_time_weekends,wake_time_weeke
  mutate(across(c(ID,frq_snore,frq_snort_or_stop_breathing,sleep_trouble,overly_sleepy,gende
  mutate(across(c(,),as.numeric))


pri_Q <- dat_2020[,c("sleep_trouble","BMI","avg_MAP","total_cholesterol","alcohol","gender",
pri_Q$sleep_trouble <- ifelse(dat_2020$sleep_trouble == "2",0,1)
```

Secondary Question 2

```
getwd()
setwd("./data/2020")
ALQ <- read_xpt("P_ALQ.XPT")
BIOPRO <- read_xpt("P_BIOPRO.XPT")
BMX <- read_xpt("P_BMX.XPT")
BPXO <- read_xpt("P_BPXO.XPT")
```

```r
DEMO <- read_xpt("P_DEMO.XPT")
PAQ <- read_xpt("P_PAQ.XPT")
SLQ <- read_xpt("P_SLQ.XPT")
TCHOL <- read_xpt("P_TCHOL.XPT")
DPQ_raw <- read_xpt("P_DPQ.XPT")

#secondary question datasets
GLU <- read_xpt("P_GLU.XPT")
INS <- read_xpt("P_INS.XPT")
PERNT <- read_xpt("P_PERNT.XPT")
UIO <- read_xpt("P_UIO.XPT")
TRIGLY <- read_xpt("P_TRIGLY.XPT")
TCHOL <- read_xpt("P_TCHOL.XPT")
HUQ <- read_xpt("P_HUQ.XPT")

DPQ <- DPQ_raw %>%
    filter(complete.cases(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_score = rowSums(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_category = case_when(
        depression_score <= 4 ~ "Minimal",
        depression_score <= 9 ~ "Mild",
        depression_score <= 14 ~ "Moderate",
        depression_score <= 19 ~ "Moderately severe",
        depression_score >= 20 ~ "Severe"
    ))

BPXO <- BPXO %>%
  mutate(
    MAP1 = (1 / 3) * BPXOSY1 + (2 / 3) * BPXODI1,
    MAP2 = (1 / 3) * BPXOSY2 + (2 / 3) * BPXODI2,
    MAP3 = (1 / 3) * BPXOSY3 + (2 / 3) * BPXODI3,
    avg_MAP = rowMeans(cbind(MAP1, MAP2, MAP3), na.rm = TRUE)
  )

datasets <- list(ALQ, BIOPRO, BMX, BPXO, DEMO,  PAQ, SLQ, TCHOL, DPQ)
test <- reduce(datasets, inner_join, by = "SEQN")

temp <- select(test, SEQN , BMXBMI, ALQ130, RIAGENDR, RIDRETH3, RIDAGEYR, INDFMPIR, LBXSNASI

dat_raw <- left_join(SLQ, temp)

# rename cols
```

```r
new_colnames <-  c(
  "ID",      # SEQN;Respondent_sequence_number
  "sleep_time_weekdays",       # SLQ300;Usual_sleep_time_weekdays
  "wake_time_weekdays",        # SLQ310;Usual_wake_time_weekdays
  "sleep_hours_weekdays",            # SLD012;Sleep_hours_weekdays
  "sleep_time_weekends",       # SLQ320;Usual_sleep_time_weekends
  "wake_time_weekends",        # SLQ330;Usual_wake_time_weekends
  "sleep_hours_weekends",            # SLD013;Sleep_hours_weekends
  "frq_snore",                 # SLQ030;How_often_snore
  "frq_snort_or_stop_breathing",# SLQ040;How_often_snort_or_stop_breathing
  "sleep_trouble",  # SLQ050;Ever_told_doctor_sleep_trouble
  "overly_sleepy",          # SLQ120;Feel_overly_sleepy_day
  "BMI",                  # BMXBMI;Body_Mass_Index
  "alcohol",         # ALQ121;How_often_drink_alcohol
  "gender",                           # RIAGENDR;Gender
  "race_ethnicity",            # RIDRETH1;race_ethnicity
  "age",                 # RIDAGEYR;Age_at_screening
  "income",  # INDFMPIR;income
  "Na",                    # LBXSNASI;Sodium_mmol_L
  "K",                # LBXSLSI;Potassium_mmol_L
  "Ca",             # LBDSCASI;Total_Calcium_mmol_L
  "Cl",                # LBXSCLSI;Chloride_mmol_L
  "sedentary_activity",       # PAD680;Minutes_sedentary_activity
  "dp_score",
  "dp_cate"

)
colnames(dat_raw) <- new_colnames

# remove all lines with NA
dat_raw[dat_raw == ""] <- NA
dat_clean <- dat_raw |>
  drop_na() |>
   filter(alcohol < 16 & frq_snore != 7 & frq_snore != 9 & frq_snort_or_stop_breathing != 7 &
  mutate(across(c(sleep_time_weekdays,wake_time_weekdays,sleep_time_weekends,wake_time_weeke
  mutate(across(c(ID,frq_snore,frq_snort_or_stop_breathing,sleep_trouble,overly_sleepy,gende
  mutate(across(c(,),as.numeric)) |>
  mutate(sleep_hours_avg = 2/7*sleep_hours_weekends+5/7*sleep_hours_weekdays)

sec_Q2_raw <- dat_clean[,c("ID", "sleep_trouble","sleep_hours_avg","BMI","alcohol","gender",

GLU <- GLU |>
```

```
  mutate(ID = as.factor(SEQN)) |>
  select(-SEQN)

INS <- INS |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

PERNT <- PERNT |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

UIO <- UIO |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

TRIGLY <- TRIGLY |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

HUQ <- HUQ |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

BIOPRO2 <- BIOPRO |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

TCHOL <- TCHOL |>
  mutate(ID = as.factor(SEQN))|>
  select(-SEQN)

#combine all datasets

extra_data_list <- list(sec_Q2_raw, GLU, INS, PERNT, UIO, TRIGLY, HUQ, TCHOL, BIOPRO2)

sec_Q2 <- Reduce(function(x, y) left_join(x, y, by = "ID"), extra_data_list) #!change name fi
```

13_14

Secondary Question 1

```
getwd()
setwd("./data/13_14")
ALQ <- read_xpt("ALQ_H.XPT")
BIOPRO <- read_xpt("BIOPRO_H.XPT")
BMX <- read_xpt("BMX_H.XPT")
BPX <- read_xpt("BPX_H.XPT")
CAFE <- read_xpt("CAFE_H.XPT")
CUSEZN <- read_xpt("CUSEZN_H.XPT") #zinc
DEMO <- read_xpt("DEMO_H.XPT")
DPQ_raw <- read_xpt("DPQ_H.XPT")
GLU <- read_xpt("GLU_H.XPT")
INQ <- read_xpt("INQ_H.XPT")
PAQ <- read_xpt("PAQ_H.XPT")
SLQ <- read_xpt("SLQ_H.XPT")
TCHOL <- read_xpt("TCHOL_H.XPT")
VID <- read_xpt("VID_H.XPT")
VITB12 <- read_xpt("VITB12_H.XPT")

DPQ <- DPQ_raw %>%
    filter(complete.cases(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_score = rowSums(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_category = case_when(
        depression_score <= 4 ~ "Minimal",
        depression_score <= 9 ~ "Mild",
        depression_score <= 14 ~ "Moderate",
        depression_score <= 19 ~ "Moderately severe",
        depression_score >= 20 ~ "Severe"
    ))

BPX <- BPX %>%
  mutate(
    avg_systolic = rowMeans(select(., BPXSY1, BPXSY2, BPXSY3, BPXSY4), na.rm = TRUE),
    avg_diastolic = rowMeans(select(., BPXDI1, BPXDI2, BPXDI3, BPXDI4), na.rm = TRUE),
    avg_MAP = 2/3*avg_diastolic+ 1/3*avg_systolic
  )

PAQ <- PAQ %>%
```

```
  mutate(
    PAQ710 = case_when(
      PAQ710 == 8 ~ 0,
      PAQ710 == 77 ~ NA_real_,
      PAQ710 == 99 ~ NA_real_,
      TRUE ~ as.numeric(PAQ710)
    ),
    PAQ715 = case_when(
      PAQ715 == 8 ~ 0,
      PAQ715 == 77 ~ NA_real_,
      PAQ715 == 99 ~ NA_real_,
      TRUE ~ as.numeric(PAQ715)
    ),
    screen_time = PAQ710 + PAQ715
  )



datasets <- list(ALQ, BIOPRO, BMX,BPX, DEMO , PAQ, SLQ, TCHOL, DPQ , INQ, VID, VITB12)
test <- reduce(datasets, inner_join, by = "SEQN")


temp <- select(test, SEQN, RIDAGEYR, RIAGENDR, RIDRETH3, ALQ120Q , BMXBMI, avg_MAP,INDFMMPI,

dat_raw_13 <- left_join(SLQ, temp)

# rename cols
new_colnames <- c(

  "ID",                           # SEQN; Respondent sequence number
  "sleep_hours",             # SLD010H; Usual hours of sleep on weekdays
  "sleep_trouble",           # SLQ050; Ever told doctor had trouble sleeping?
  "sleep_diagnosed",         # SLQ060; Ever told by doctor have sleep disorder?
  "age",                          # RIDAGEYR; Age at screening
  "gender",                       # RIAGENDR; Gender
  "race_ethnicity",               # RIDRETH3; Race/Ethnicity category
  "alcohol",     # ALQ120Q; How often drank alcohol in past 12 months
  "BMI",
  "avg_MAP",               # avg_MAP
  "income", # INDFMMPI; Family income to poverty ratio
  "screen_time",                   # TV+computer time
  "sedentary_minutes",            # PAD680; Minutes of sedentary activity
```

```
  "total_cholesterol",            # LBDTCSI; Total cholesterol (mg/dL)
  "depression_score",             # depression_score; Calculated total depression score
  "depression_category",          # depression_category; Depression score category
  "vitamin_d",                    # LBXVIDMS; Vitamin D level (nmol/L)
  "vitamin_b12"                   # LBDB12SI; Vitamin B12 level (pmol/L)
)
colnames(dat_raw_13) <- new_colnames
# remove all lines with NA
dat_raw_13[dat_raw_13 == ""] <- NA
dat_13 <- dat_raw_13 |>
  drop_na() |>
  filter(alcohol < 400  & sedentary_minutes < 6000 & sleep_hours < 66 & sleep_trouble < 3 & s
  mutate(across(c(ID, sleep_trouble, sleep_diagnosed, gender, race_ethnicity), as.factor))



sec_Q1 <- dat_13[,c("sleep_trouble","BMI","alcohol","gender","race_ethnicity","age","income"
sec_Q1$sleep_trouble <- ifelse(dat_13$sleep_trouble == "2",0,1)

sec_Q3_2 <- dat_13[,c("sleep_diagnosed","BMI","alcohol","gender","race_ethnicity","age","inco
sec_Q3_2$sleep_diagnosed <- ifelse(dat_13$sleep_diagnosed == "2",0,1)
```

15-16

```
getwd()
setwd("./data/15_16")
ALQ <- read_xpt("ALQ_I.xpt")
BIOPRO <- read_xpt("BIOPRO_I.xpt")
BMX <- read_xpt("BMX_I.xpt")
BPXO <- read_xpt("BPX_I.xpt")
DEMO <- read_xpt("DEMO_I.xpt")
DPQ_raw <- read_xpt("DPQ_I.xpt")
GLU <- read_xpt("GLU_I.xpt")
HUQ <- read_xpt("HUQ_I.xpt")
INS <- read_xpt("INS_I.xpt")
PAQ <- read_xpt("PAQ_I.xpt")
PERNT <- read_xpt("PERNT_I.xpt")
SLQ <- read_xpt("SLQ_I.xpt")
TCHOL <- read_xpt("TCHOL_I.xpt")
TRIGLY <- read_xpt("TRIGLY_I.xpt")
```

```r
UIO <- read_xpt("UIO_I.xpt")

DPQ <- DPQ_raw %>%
    filter(complete.cases(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_score = rowSums(select(., starts_with("DPQ"))[-1])) |> #%>%
    mutate(depression_category = case_when(
        depression_score <= 4 ~ "Minimal",
        depression_score <= 9 ~ "Mild",
        depression_score <= 14 ~ "Moderate",
        depression_score <= 19 ~ "Moderately severe",
        depression_score >= 20 ~ "Severe"
    ))

BPXO <- BPXO %>%
  mutate(
    MAP1 = (1 / 3) * BPXSY1 + (2 / 3) * BPXDI1,
    MAP2 = (1 / 3) * BPXSY2 + (2 / 3) * BPXDI2,
    MAP3 = (1 / 3) * BPXSY3 + (2 / 3) * BPXDI3,
    avg_MAP = rowMeans(cbind(MAP1, MAP2, MAP3), na.rm = TRUE)
  )


datasets <- list(ALQ, BIOPRO, BMX, BPXO, DEMO,  PAQ, SLQ, TCHOL, DPQ)
test <- reduce(datasets, inner_join, by = "SEQN")
temp <- select(test, SEQN , BMXBMI, avg_MAP, ALQ130, RIAGENDR, RIDRETH3, RIDAGEYR, INDFMPIR,

dat_raw_15 <- left_join(SLQ, temp)

new_colnames <-  c(
  "ID",                          # SEQN; Respondent sequence number
  "sleep_time_weekdays",       # SLQ300;Usual_sleep_time_weekdays
  "wake_time_weekdays",        # SLQ310;Usual_wake_time_weekdays
  "sleep_hours",             # SLD012;Sleep_hours
  "frq_snore",                 # SLQ030;How_often_snore
  "frq_snort_or_stop_breathing",# SLQ040;How_often_snort_or_stop_breathing
  "sleep_trouble",  # SLQ050;Ever_told_doctor_sleep_trouble
  "overly_sleepy",           # SLQ120;Feel_overly_sleepy_day
  "BMI",                         # BMXBMI; Body Mass Index
  "avg_MAP",                     # avg_MAP; Average Mean Arterial Pressure
  "alcohol",                     # ALQ130; How often drink alcohol
  "gender",                      # RIAGENDR; Gender
  "race_ethnicity",         # RIDRETH1; Race/Hispanic origin
```

```
  "age",                          # RIDAGEYR; Age at screening
  "income",  # INDFMPIR; Family income to poverty ratio
  "sedentary_activity",           # PAD680; Minutes sedentary activity
  "total_cholesterol",            # LBDTCSI; Total cholesterol mmol/L
  "depression_score",             # depression_score; Depression score
  "depression_category"           # depression_category; Depression category
)
colnames(dat_raw_15) <- new_colnames
#remove all lines with NA
dat_raw_15[dat_raw_15 == ""] <- NA
dat_1516 <- dat_raw_15 |>
  drop_na() |>
   filter(alcohol < 16 & frq_snore != 7 & frq_snore != 9 & frq_snort_or_stop_breathing != 7 &
  mutate(across(c(ID,frq_snore,frq_snort_or_stop_breathing,sleep_trouble,overly_sleepy,gender
  mutate(across(c(,),as.numeric))

sec_Q3 <- dat_1516[,c("sleep_trouble","BMI","avg_MAP","total_cholesterol","alcohol","gender"
sec_Q3$sleep_trouble <- ifelse(dat_1516$sleep_trouble == "2",0,1)
```

EDA

2020

```
summary(dat_2020)
numeric_vars <- select(dat_2020, age, BMI, avg_MAP, income,
                       sedentary_activity, total_cholesterol, sleep_hours_weekdays, depressio
for (col in colnames(numeric_vars)) {
  data_range <- range(dat_2020[[col]], na.rm = TRUE)
  binwidth <- (data_range[2] - data_range[1]) / 30

  p <- ggplot(dat_2020, aes_string(x = col)) +
    geom_histogram(binwidth = binwidth, alpha = 0.7) +
    labs(title = paste("Distribution of", col), x = col, y = "Frequency") +
    theme_minimal()
  print(p)
}
categorical_vars <- select(dat_2020, gender, race_ethnicity, sleep_trouble, overly_sleepy, de

for (col in colnames(categorical_vars)) {
```

```
  p <- ggplot(dat_2020, aes_string(x = col)) +
    geom_bar( alpha = 0.7) +
    labs(title = paste("Distribution of", col), x = col, y = "Count") +
    theme_minimal() +
    scale_x_discrete(drop = FALSE)
  print(p)
}
```

```
for (col in colnames(numeric_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_2020, aes_string(x = col, y = "depression_score")) +
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs Depression Score"), x = col, y = "Depres
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_category") {
    p <- ggplot(dat_2020, aes_string(x = col, y = "depression_score")) +
      geom_boxplot(fill = "purple", alpha = 0.7) +
      labs(title = paste("Box Plot of", col, "vs Depression Score"), x = col, y = "Depressio
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_category") {
    p <- ggplot(dat_2020, aes_string(x = col, fill = "depression_category")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Depression Category"), x = col, y = "Pr
      theme_minimal()
    print(p)
  }
}
```

```
for (col in colnames(numeric_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_2020, aes_string(x = col, y = "depression_score")) +
```

```
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs Depression Score"), x = col, y = "Depres
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_2020, aes_string(x = col, y = "depression_score")) +
      geom_boxplot(fill = "purple", alpha = 0.7) +
      labs(title = paste("Box Plot of", col, "vs Depression Score"), x = col, y = "Depressio
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_2020, aes_string(x = col, fill = "depression_category")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Depression Category"), x = col, y = "Pr
      theme_minimal()
    print(p)
  }
}
```

```
for (col in colnames(numeric_vars)) {
  p <- ggplot(dat_2020, aes_string(x = "sleep_trouble", y = col)) +
    geom_boxplot(fill = "purple", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs Sleep Trouble"), x = "Sleep Trouble", y = col
    theme_minimal()
  print(p)
}


for (col in colnames(categorical_vars)) {
  if (col != "sleep_trouble") {
    p <- ggplot(dat_2020, aes_string(x = col, fill = "sleep_trouble")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Sleep Trouble"), x = col, y = "Proportio
```

```
      theme_minimal() +
      scale_fill_discrete(drop = FALSE)  # Ensure all levels of sleep_trouble are represented
    print(p)
  }
}
```

```
for (col in colnames(numeric_vars)) {
  p <- ggplot(dat_2020, aes_string(x = "frq_snore", y = col)) +
    geom_boxplot(fill = "orange", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs frq_snore"), x = "frq_snore", y = col) +
    theme_minimal()
  print(p)
}

for (col in colnames(categorical_vars)) {
  if (col != "sleep_diagnosed") {
    p <- ggplot(dat_2020, aes_string(x = col, fill = "frq_snore")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by frq_snore"), x = col, y = "Proportion") +
      theme_minimal() +
      scale_fill_discrete(drop = FALSE)
    print(p)
  }
}
```

```
for (col in colnames(numeric_vars)) {
  if (col != "sleep_hours_weekdays") {
    p <- ggplot(dat_2020, aes_string(x = col, y = "sleep_hours_weekdays")) +
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs sleep_hours_weekdays"), x = col, y = "sl
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  p <- ggplot(dat_2020, aes_string(x = col, y = "sleep_hours_weekdays")) +
    geom_boxplot(fill = "cyan", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs sleep_hours_weekdays"), x = col, y = "sleep_ho
    theme_minimal()
  print(p)
```

```
}
```

```
ggplot(dat_2020, aes(x = sleep_trouble, fill = frq_snore)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(
    title = "Relationship between Sleep Trouble  and frq_snore ",
    x = "Told Doctor They Had Sleep Trouble",
    y = "Count",
    fill = "Diagnosed by Doctor"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")
```

```
ggplot(dat_2020, aes(x = sleep_trouble, y = sleep_hours_weekdays)) +
  geom_boxplot(fill = "skyblue", alpha = 0.7) +
  labs(
    title = "Relationship between Sleep Trouble (Self-Reported) and sleep hours",
    x = "Sleep Trouble (Self-Reported)",
    y = "Usual Sleep Hours"
  ) +
  theme_minimal()
```

```
ggplot(dat_2020, aes(x = frq_snore, y = sleep_hours_weekdays)) +
  geom_boxplot(fill = "skyblue", alpha = 0.7) +
  labs(
    title = "Relationship between frq_snore and Sleep Hours",
    x = "frq_snore",
    y = "Usual Sleep Hours"
  ) +
  theme_minimal()
```

13_14

Correlations

```
summary(dat_13)

numeric_vars <- select(dat_13, age, BMI, avg_MAP, income, screen_time, sedentary_minutes, to
```

```r
for (col in colnames(numeric_vars)) {
  data_range <- range(dat_13[[col]], na.rm = TRUE)
  binwidth <- (data_range[2] - data_range[1]) / 30

  p <- ggplot(dat_13, aes_string(x = col)) +
    geom_histogram(binwidth = binwidth, alpha = 0.7) +
    labs(title = paste("Distribution of", col), x = col, y = "Frequency") +
    theme_minimal()
  print(p)
}


categorical_vars <- select(dat_13, gender, race_ethnicity, sleep_trouble, sleep_diagnosed, de

for (col in colnames(categorical_vars)) {
  p <- ggplot(dat_13, aes_string(x = col)) +
    geom_bar( alpha = 0.7) +
    labs(title = paste("Distribution of", col), x = col, y = "Count") +
    theme_minimal() +
    scale_x_discrete(drop = FALSE)
  print(p)
}
```

```r
for (col in colnames(numeric_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_13, aes_string(x = col, y = "depression_score")) +
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs Depression Score"), x = col, y = "Depres
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_category") {
    p <- ggplot(dat_13, aes_string(x = col, y = "depression_score")) +
      geom_boxplot(fill = "purple", alpha = 0.7) +
      labs(title = paste("Box Plot of", col, "vs Depression Score"), x = col, y = "Depressio
      theme_minimal()
    print(p)
  }
```

```
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_category") {
    p <- ggplot(dat_13, aes_string(x = col, fill = "depression_category")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Depression Category"), x = col, y = "Pro
      theme_minimal()
    print(p)
  }
}
```

```
for (col in colnames(numeric_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_13, aes_string(x = col, y = "depression_score")) +
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs Depression Score"), x = col, y = "Depres
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_13, aes_string(x = col, y = "depression_score")) +
      geom_boxplot(fill = "purple", alpha = 0.7) +
      labs(title = paste("Box Plot of", col, "vs Depression Score"), x = col, y = "Depressio
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  if (col != "depression_score") {
    p <- ggplot(dat_13, aes_string(x = col, fill = "depression_category")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Depression Category"), x = col, y = "Pro
      theme_minimal()
    print(p)
  }
}
```

```r
for (col in colnames(numeric_vars)) {
  p <- ggplot(dat_13, aes_string(x = "sleep_trouble", y = col)) +
    geom_boxplot(fill = "purple", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs Sleep Trouble"), x = "Sleep Trouble", y = col)
    theme_minimal()
  print(p)
}


for (col in colnames(categorical_vars)) {
  if (col != "sleep_trouble") {
    p <- ggplot(dat_13, aes_string(x = col, fill = "sleep_trouble")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Sleep Trouble"), x = col, y = "Proportio
      theme_minimal() +
      scale_fill_discrete(drop = FALSE)  # Ensure all levels of sleep_trouble are represented
    print(p)
  }
}
```

```r
for (col in colnames(numeric_vars)) {
  p <- ggplot(dat_13, aes_string(x = "sleep_diagnosed", y = col)) +
    geom_boxplot(fill = "orange", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs Sleep Diagnosed"), x = "Sleep Diagnosed", y =
    theme_minimal()
  print(p)
}

for (col in colnames(categorical_vars)) {
  if (col != "sleep_diagnosed") {
    p <- ggplot(dat_13, aes_string(x = col, fill = "sleep_diagnosed")) +
      geom_bar(position = "fill") +
      labs(title = paste("Distribution of", col, "by Sleep Diagnosed"), x = col, y = "Proport
      theme_minimal() +
      scale_fill_discrete(drop = FALSE)
    print(p)
  }
}
```

```r
for (col in colnames(numeric_vars)) {
  if (col != "sleep_hours") {
    p <- ggplot(dat_13, aes_string(x = col, y = "sleep_hours")) +
```

```
      geom_point(alpha = 0.5) +
      geom_smooth(method = "lm", col = "red", se = FALSE) +
      labs(title = paste("Scatter Plot of", col, "vs Sleep Hours"), x = col, y = "Sleep Hours
      theme_minimal()
    print(p)
  }
}

for (col in colnames(categorical_vars)) {
  p <- ggplot(dat_13, aes_string(x = col, y = "sleep_hours")) +
    geom_boxplot(fill = "cyan", alpha = 0.7) +
    labs(title = paste("Box Plot of", col, "vs Sleep Hours"), x = col, y = "Sleep Hours") +
    theme_minimal()
  print(p)
}
```

```
ggplot(dat_13, aes(x = sleep_trouble, fill = sleep_diagnosed)) +
  geom_bar(position = "dodge", alpha = 0.8) +
  labs(
    title = "Relationship between Sleep Trouble (Self-Reported) and Sleep Diagnosed (By Docto
    x = "Told Doctor They Had Sleep Trouble",
    y = "Count",
    fill = "Diagnosed by Doctor"
  ) +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")
```

```
ggplot(dat_13, aes(x = sleep_trouble, y = sleep_hours)) +
  geom_boxplot(fill = "skyblue", alpha = 0.7) +
  labs(
    title = "Relationship between Sleep Trouble (Self-Reported) and Sleep Hours",
    x = "Sleep Trouble (Self-Reported)",
    y = "Usual Sleep Hours"
  ) +
  theme_minimal()
```

```
ggplot(dat_13, aes(x = sleep_diagnosed, y = sleep_hours)) +
  geom_boxplot(fill = "skyblue", alpha = 0.7) +
  labs(
    title = "Relationship between Sleep Diagnosed and Sleep Hours",
    x = "Sleep Trouble (Self-Reported)",
```

```
    y = "Usual Sleep Hours"
  ) +
  theme_minimal()
```

Primary Question

categorical depression_score vs continuous depression_score

```
cate <- glm(sleep_trouble ~ as.factor(depression_category), data = pri_Q, family = "binomial"
conti <- glm(sleep_trouble ~ depression_score, data = pri_Q, family = "binomial")
summary(cate)
summary(conti)
anova(cate, conti, test="Chisq")
pri_Q <- pri_Q |>
  select(-depression_category)
```

VIF

```
full_model <- glm(sleep_trouble ~ BMI+alcohol+gender+race_ethnicity+age+income+depression_sc
vif(full_model)
```

Check the assumption that the relationship between a categorical covariate and an outcome changes linearly from one category to the next !assumptions

```
plot(conti$fitted.values~pri_Q$depression_score,type='p',
     col='black',ylab="P(sleep~disorder)",xlab="age")
lines(cate$fitted.values~pri_Q$depression_score,type='p',
      col="red")
legend('bottomright',legend=c("linear","ordinal"),
       pch=c(1,1),col=c(1,2))
```

potential confounders

BMI, alcohol, gender, race_ethnicity, age, and income appear to satisfy the causal definition of a confounder

```
mod2 <- glm(sleep_trouble ~ BMI+alcohol+gender+race_ethnicity+age+income+depression_score+avg
summary(mod2)
```

potential effect modifier

```
mod3 <- update(mod2,. ~ .+depression_score*BMI)
coef(summary(mod3))["BMI:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*alcohol)
coef(summary(mod3))["alcohol:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*gender)
coef(summary(mod3))["gender2:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*race_ethnicity)
summary(mod3)
mod3 <- update(mod2,. ~ .+depression_score*age)
coef(summary(mod3))["age:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*income)
summary(mod3)
mod3 <- update(mod2,. ~ .+depression_score*avg_MAP)
coef(summary(mod3))["depression_score:avg_MAP", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*total_cholesterol)
coef(summary(mod3))["depression_score:total_cholesterol", "Pr(>|z|)"]
```

Assess possible nonlinear effect of BMI, alcohol, income, and age

```
mod3 <- update(mod2,. ~ .+ I(BMI^2))
coef(summary(mod3))["I(BMI^2)", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+ I(alcohol^2))
coef(summary(mod3))["I(alcohol^2)", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+ I(age^2))
coef(summary(mod3))["I(age^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+ I(alcohol^2))
coef(summary(mod4))["I(alcohol^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+ I(income^2))
```

```
coef(summary(mod4))["I(income^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+I(avg_MAP^2))
coef(summary(mod4))["I(avg_MAP^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+I(total_cholesterol^2))
coef(summary(mod4))["I(total_cholesterol^2)", "Pr(>|z|)"]
```

check high influence points

```
par(mfrow=c(1,1))
influencePlot(mod3,col="red")
influenceIndexPlot(mod3)
```

GOF

Hosmer-Lemeshow test because we have a larger number of covariate patterns.

```
library(ResourceSelection)
options(digits=7)
# Hosmer-Lemeshow Test
hoslem.test(mod3$y,fitted(mod3),g=10)
```

ROC

```
library(pROC)
predprob <- predict(mod3,type=c("response"))
roccurve <- roc(sleep_trouble ~ predprob, data = pri_Q)
plot(roccurve,col="red")
auc(roccurve)
```

Accuracy

```
final_model_pri <- mod3
predprob <- predict(final_model_pri,type=c("response"))
predicted <- ifelse(predprob > 0.5, 1, 0)
predicted <- factor(predicted, levels = c(0, 1))
```

```
actual <- pri_Q$sleep_trouble
actual <- factor(actual, levels = c(0, 1))
confusionMatrix(predicted, actual)$overall["Accuracy"]
```

Secondary Question

Secondary Question 1

categorical depression_score vs continuous depression_score

```
cate <- glm(sleep_trouble ~ as.factor(depression_category), data = sec_Q1, family = "binomial
conti <- glm(sleep_trouble ~ depression_score, data = sec_Q1, family = "binomial")
summary(cate)
summary(conti)
anova(cate, conti, test="Chisq")
sec_Q1 <- sec_Q1 |>
  select(-depression_category)
```

VIF

```
full_model <- glm(sleep_trouble ~ BMI+alcohol+gender+race_ethnicity+age+income+depression_sc
vif(full_model)
```

Check the assumption that the relationship between a categorical covariate and an outcome changes linearly from one category to the next

```
plot(conti$fitted.values~sec_Q1$depression_score,type='p',
     col='black',ylab="P(sleep~disorder)",xlab="age")
lines(cate$fitted.values~sec_Q1$depression_score,type='p',
      col="red")
legend('bottomright',legend=c("linear","ordinal"),
       pch=c(1,1),col=c(1,2))
```

potential confounders

BMI, alcohol, gender, race_ethnicity, age, and income appear to satisfy the causal definition of a confounder

```
mod2 <- glm(sleep_trouble ~ BMI+alcohol+gender+race_ethnicity+age+income+depression_score+avg
summary(mod2)
```

potential effect modifier

```
mod3 <- update(mod2,. ~ .+depression_score*BMI)
coef(summary(mod3))["BMI:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*alcohol)
coef(summary(mod3))["alcohol:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*gender)
coef(summary(mod3))["gender2:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*race_ethnicity)
summary(mod3)
mod3 <- update(mod2,. ~ .+depression_score*age)
coef(summary(mod3))["age:depression_score", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+depression_score*income)
summary(mod3)
```

Assess possible nonlinear effect of BMI, alcohol, income, and age

```
mod3 <- update(mod2,. ~ .+ I(BMI^2))
coef(summary(mod3))["I(BMI^2)", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+ I(alcohol^2))
coef(summary(mod3))["I(alcohol^2)", "Pr(>|z|)"]
mod3 <- update(mod2,. ~ .+ I(age^2))
coef(summary(mod3))["I(age^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+ I(alcohol^2))
coef(summary(mod4))["I(alcohol^2)", "Pr(>|z|)"]
mod4 <- update(mod3,. ~ .+ I(income^2))
coef(summary(mod4))["I(income^2)", "Pr(>|z|)"]
summary(mod3)
```

check high influence points

```
par(mfrow=c(1,1))
influencePlot(mod3,col="red")
influenceIndexPlot(mod3)
```

GOF

Hosmer-Lemeshow test because we have a larger number of covariate patterns.

```
library(ResourceSelection)
options(digits=7)
# Hosmer-Lemeshow Test
hoslem.test(mod3$y,fitted(mod3),g=10)
```

ROC

```
library(pROC)
final_model_sec1 <- mod3
predprob <- predict(mod3,type=c("response"))
roccurve <- roc(sleep_trouble ~ predprob, data = sec_Q1)
plot(roccurve,col="red")
auc(roccurve)
```

Secondary Question 2

Association Analysis for Sleep Hours

```
library(glmnet)

## Electrolytes

# Sleep hours and phosphorous
lm_phs = lm(sleep_hours_avg ~ LBDSPHSI, data = sec_Q2)
summary(lm_phs) #significant

# Sleep hours and sodium
```

```r
lm_sod = lm(sleep_hours_avg ~ LBXSNASI, data = sec_Q2)
summary(lm_sod)

# Sleep hours and potassium
lm_k = lm(sleep_hours_avg ~ LBXSKSI, data = sec_Q2)
summary(lm_k)

# Sleep hours and chloride
lm_cl = lm(sleep_hours_avg ~ LBXSCLSI, data = sec_Q2)
summary(lm_cl)

# Sleep hours and calcium
lm_ca = lm(sleep_hours_avg ~ LBDSCASI, data = sec_Q2)
summary(lm_ca)

# Sleep hours and iodine
lm_iod = lm(sleep_hours_avg ~ URXUIO, data = sec_Q2)
summary(lm_iod)

# Sleep hours and iron
lm_fe = lm(sleep_hours_avg ~ LBDSIRSI, data = sec_Q2)
summary(lm_fe)


## Common Biochemicals

# Sleep hours and insulin
lm_ins = lm(sleep_hours_avg ~ LBDINSI, data = sec_Q2)
summary(lm_ins)

# Uric Acid
lm_ura = lm(sleep_hours_avg ~ LBXSUA, data = sec_Q2)
summary(lm_ura) #highly significant

# Blood glucose
lm_glu = lm(sleep_hours_avg ~ LBDSGLSI, data = sec_Q2)
summary(lm_glu) #significant

# Nitrate
lm_nit = lm(sleep_hours_avg ~ URXNO3, data = sec_Q2)
summary(lm_nit)
```

```r
# Total cholesterol
lm_tch = lm(sleep_hours_avg ~ LBDTCSI, data = sec_Q2)
summary(lm_tch)

## Main covariates from primary analysis

#depression
lm_dpscore = lm(sleep_hours_avg ~ dp_score, data = sec_Q2)
summary(lm_dpscore)

lm_dpcat = lm(sleep_hours_avg ~ dp_cate, data = sec_Q2)
summary(lm_dpcat)


## Stepwise Model Selection

filtered_dat <- sec_Q2 %>%
  filter(!is.na(sleep_hours_avg) &
           !is.na(LBDSPHSI) & !is.na(LBXSNASI) & !is.na(LBXSKSI) &
           !is.na(LBXSCLSI) & !is.na(LBDSCASI) & !is.na(URXUIO) &
           !is.na(LBDSIRSI) & !is.na(LBDINSI) & !is.na(LBXSUA) &
           !is.na(LBDSGLSI) & !is.na(URXNO3) & !is.na(LBDTCSI) &
           !is.na(dp_score) & !is.na(dp_cate))


lm_step <- lm(sleep_hours_avg ~ LBDSPHSI + LBXSNASI + LBXSKSI + LBXSCLSI + LBDSCASI +
                URXUIO + LBDSIRSI + LBDINSI + LBXSUA + LBDSGLSI + URXNO3 + LBDTCSI +
                dp_score + dp_cate, data=filtered_dat)
summary(lm_step)
stepModel <- step(lm_step, direction=c("both"))
summary(stepModel)
plot(fitted(stepModel), residuals(stepModel))
abline(a=0,b=0,col="pink")

qqnorm(residuals(stepModel))
qqline(residuals(stepModel),col="pink")

# Elastic Net
x <- model.matrix(~ LBDSPHSI + LBXSNASI + LBXSKSI + LBXSCLSI +
                    LBDSCASI + LBDSIRSI + LBXSUA + LBDSGLSI +
                    LBDTCSI + dp_score + BMI + gender +
                    race_ethnicity + income + age, data = sec_Q2)[,-1]
```

```r
y <- sec_Q2$sleep_hours_avg

lambda_grid <- 10^seq(3, -3, length = 100)

EN_model <- glmnet(x, y, alpha = 0.5, lambda = lambda_grid)

set.seed(123)
cv.EN <- cv.glmnet(x, y, alpha = 0.5, lambda = lambda_grid)

lambda_min_EN <- cv.EN$lambda.min
lambda_1se_EN <- cv.EN$lambda.1se

print(paste("Best lambda (min):", lambda_min_EN))
print(paste("Best lambda (1se):", lambda_1se_EN))

plot(cv.EN)

final_model <- glmnet(x, y, alpha = 0.5, lambda = lambda_min_EN)

coef(final_model)

fitted_values <- predict(final_model, newx = x, s = lambda_min_EN)
fitted_values <- as.vector(fitted_values)
residuals_values <- y - fitted_values

plot(fitted_values, residuals_values,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residual Plot")
abline(h = 0, col = "pink")

qqnorm(residuals_values, main = "Q-Q Plot of Residuals")
qqline(residuals_values, col = "blue")
```

Secondary Question 3

predict 15-16 by primary-Q model

```r
predprob <- predict(object = final_model_pri, newdata = sec_Q3, type = "response")
predicted <- ifelse(predprob > 0.5, 1, 0)
predicted <- factor(predicted, levels = c(0, 1))
```

```
actual <- sec_Q3$sleep_trouble
actual <- factor(actual, levels = c(0, 1))
confusionMatrix(predicted, actual)$overall["Accuracy"]
```

```
roccurve <- roc(sleep_trouble ~ predprob, data = sec_Q3)
plot(roccurve,col="red")
auc(roccurve)
```

predict 15-16 by RF

```
x <- as.matrix(pri_Q[, -1])
x <- matrix(as.numeric(x), nrow = nrow(x), ncol = ncol(x))
y <- factor(pri_Q$sleep_trouble)
x_test <- as.matrix(sec_Q3[, c(-1,-11)])
x_test <- matrix(as.numeric(x_test), nrow = nrow(x_test), ncol = ncol(x_test))
y_test <- factor(sec_Q3$sleep_trouble)
colnames(x) <- 1:ncol(as.matrix(pri_Q[, -1]))
colnames(x_test) <- colnames(x)
```

Setup parallel

```
nc <- detectCores() - 1
cl <- makeCluster(nc)
registerDoParallel(cl)
```

Random Forest

```
trees <- 300
train_rf <- train(x, y, method = "rf",
                  preProcess = "nzv",
                  tuneGrid = data.frame(mtry = seq(5, 15)),
                  ntree = trees
                  )
y_hat_rf <- predict(train_rf, x_test, type = "raw")
confusionMatrix(y_hat_rf, y_test)$overall["Accuracy"]
stopCluster(cl)
stopImplicitCluster()
```

```
y_hat_rf_prob <- predict(train_rf, x_test, type = "prob")[, 2]
roccurve <- roc(y_test ~ y_hat_rf_prob)
plot(roccurve,col="red")
auc(roccurve)
```

predict 13-14 by primary-Q model

```
predprob <- predict(object = final_model_pri, newdata = sec_Q3_2 , type = "response")
predicted <- ifelse(predprob > 0.5, 1, 0)
predicted <- factor(predicted, levels = c(0, 1))
actual <- sec_Q3_2$sleep_diagnosed
actual <- factor(actual, levels = c(0, 1))
confusionMatrix(predicted, actual)$overall["Accuracy"]
```

```
roccurve <- roc(sleep_diagnosed ~ predprob, data = sec_Q3_2 )
plot(roccurve,col="red")
auc(roccurve)
```

predict 13-14 by RF

```
x <- as.matrix(pri_Q[, -1])
x <- matrix(as.numeric(x), nrow = nrow(x), ncol = ncol(x))
y <- factor(pri_Q$sleep_trouble)
x_test <- as.matrix(sec_Q3_2[, c(-1,-9)])
x_test <- matrix(as.numeric(x_test), nrow = nrow(x_test), ncol = ncol(x_test))
y_test <- factor(sec_Q3_2$sleep_diagnosed)
colnames(x) <- 1:ncol(as.matrix(pri_Q[, -1]))
colnames(x_test) <- colnames(x)
```

Setup parallel

```
nc <- detectCores() - 1
cl <- makeCluster(nc)
registerDoParallel(cl)
```

Random Forest

```
trees <- 300
train_rf <- train(x, y, method = "rf",
                  preProcess = "nzv",
                  tuneGrid = data.frame(mtry = seq(5, 15)),
                  ntree = trees
                  )
y_hat_rf <- predict(train_rf, x_test, type = "raw")
confusionMatrix(y_hat_rf, y_test)$overall["Accuracy"]
stopCluster(cl)
stopImplicitCluster()
```

```
y_hat_rf_prob <- predict(train_rf, x_test, type = "prob")[, 2]
roccurve <- roc(y_test ~ y_hat_rf_prob)
plot(roccurve,col="red")
auc(roccurve)
```

Sensitivity analysis

```
high_influence <-  as.numeric(rownames(influencePlot(final_model_pri,col="red")))
pri_Q[high_influence, ]
summary(pri_Q)
```

Try Remove

```
sen_dat_pri <- pri_Q[-high_influence, ]
mod_sen <- update(final_model_pri, data = sen_dat_pri)
summary(mod_sen)$aic
summary(final_model_pri)$aic
```

```
high_influence <-  as.numeric(rownames(influencePlot(final_model_sec1,col="red")))
sen_dat_pri <- pri_Q[-high_influence, ]
mod_sen <- update(final_model_sec1, data = sen_dat_pri)
summary(mod_sen)$aic
summary(final_model_sec1)$aic
```

Imputation dataset

2020

```
getwd()
setwd("./data/2020")

ALQ <- read_xpt("P_ALQ.XPT")
BIOPRO <- read_xpt("P_BIOPRO.XPT")
BMX <- read_xpt("P_BMX.XPT")
BPXO <- read_xpt("P_BPXO.XPT")
DEMO <- read_xpt("P_DEMO.XPT")
#FOLATE <- read_xpt("P_FOLATE.XPT")
PAQ <- read_xpt("P_PAQ.XPT")
SLQ <- read_xpt("P_SLQ.XPT")
TCHOL <- read_xpt("P_TCHOL.XPT")
DPQ_raw <- read_xpt("P_DPQ.XPT")
FETIB <- read_xpt("P_FETIB.XPT")

DPQ <- DPQ_raw %>%
    filter(complete.cases(select(., starts_with("DPQ"))[-1])) %>%
    mutate(depression_score = rowSums(select(., starts_with("DPQ"))[-1])) |> #%>%
    mutate(depression_category = case_when(
        depression_score <= 4 ~ "Minimal",
        depression_score <= 9 ~ "Mild",
        depression_score <= 14 ~ "Moderate",
        depression_score <= 19 ~ "Moderately severe",
        depression_score >= 20 ~ "Severe"
    ))

BPXO <- BPXO %>%
  mutate(
    MAP1 = (1 / 3) * BPXOSY1 + (2 / 3) * BPXODI1,
    MAP2 = (1 / 3) * BPXOSY2 + (2 / 3) * BPXODI2,
    MAP3 = (1 / 3) * BPXOSY3 + (2 / 3) * BPXODI3,
    avg_MAP = rowMeans(cbind(MAP1, MAP2, MAP3), na.rm = TRUE)
  )
datasets <- list(ALQ, BIOPRO, BMX, BPXO, DEMO,  PAQ, SLQ, TCHOL, DPQ)
test <- reduce(datasets, inner_join, by = "SEQN")
```

```r
temp <- select(test, SEQN , BMXBMI, avg_MAP, ALQ130, RIAGENDR, RIDRETH3, RIDAGEYR, INDFMPIR,

dat_raw <- left_join(SLQ, temp)

new_colnames <-  c(
  "ID",                          # SEQN; Respondent sequence number
  "sleep_time_weekdays",       # SLQ300;Usual_sleep_time_weekdays
  "wake_time_weekdays",        # SLQ310;Usual_wake_time_weekdays
  "sleep_hours_weekdays",           # SLD012;Sleep_hours_weekdays
  "sleep_time_weekends",       # SLQ320;Usual_sleep_time_weekends
  "wake_time_weekends",        # SLQ330;Usual_wake_time_weekends
  "sleep_hours_weekends",           # SLD013;Sleep_hours_weekends
  "frq_snore",                 # SLQ030;How_often_snore
  "frq_snort_or_stop_breathing",# SLQ040;How_often_snort_or_stop_breathing
  "sleep_trouble",  # SLQ050;Ever_told_doctor_sleep_trouble
  "overly_sleepy",          # SLQ120;Feel_overly_sleepy_day
  "BMI",                         # BMXBMI; Body Mass Index
  "avg_MAP",                     # avg_MAP; Average Mean Arterial Pressure
  "alcohol",                     # ALQ130; How often drink alcohol
  "gender",                      # RIAGENDR; Gender
  "race_ethnicity",        # RIDRETH1; Race/Hispanic origin
  "age",                         # RIDAGEYR; Age at screening
  "income",  # INDFMPIR; Family income to poverty ratio
  "sedentary_activity",          # PAD680; Minutes sedentary activity
  "total_cholesterol",           # LBDTCSI; Total cholesterol mmol/L
  "depression_score",            # depression_score; Depression score
  "depression_category"          # depression_category; Depression category
)
colnames(dat_raw) <- new_colnames
# remove all lines with NA
dat_raw[dat_raw == ""] <- NA
dat_clean <- dat_raw |>
  drop_na() |>
   filter(alcohol < 16 & frq_snore != 7 & frq_snore != 9 & frq_snort_or_stop_breathing != 7 &
  mutate(across(c(sleep_time_weekdays,wake_time_weekdays,sleep_time_weekends,wake_time_weeke
  mutate(across(c(ID,frq_snore,frq_snort_or_stop_breathing,sleep_trouble,overly_sleepy,gende
  mutate(across(c(,),as.numeric)) |>
  mutate(sleep_hours_avg = 2/7*sleep_hours_weekends+5/7*sleep_hours_weekdays)

dat_temp <- DEMO
other_dfs <- Filter(function(df) !is.null(df) && "SEQN" %in% colnames(df),
                    list(DEMO, SLQ,ALQ, BIOPRO, BMX, BPXO, DEMO,  PAQ,  TCHOL, DPQ))
```

```
for (df in other_dfs) {
    dat_temp <- left_join(dat_temp, df, by = "SEQN")
}
dat_subset <- select(dat_temp,
                     SEQN,                  # Corresponds to "ID"
                     SLQ300,                # Corresponds to "sleep_time_weekdays"
                     SLQ310,                # Corresponds to "wake_time_weekdays"
                     SLD012,                # Corresponds to "sleep_hours_weekdays"
                     SLQ320,                # Corresponds to "sleep_time_weekends"
                     SLQ330,                # Corresponds to "wake_time_weekends"
                     SLD013,                # Corresponds to "sleep_hours_weekends"
                     SLQ030,                # Corresponds to "frq_snore"
                     SLQ040,                # Corresponds to "frq_snort_or_stop_breathing"
                     SLQ050,                # Corresponds to "sleep_trouble"
                     SLQ120,                # Corresponds to "overly_sleepy"
                     BMXBMI,                # Corresponds to "BMI"
                     avg_MAP,               # Corresponds to "avg_MAP"
                     ALQ130,                # Corresponds to "alcohol"
                     RIAGENDR,              # Corresponds to "gender"
                     RIDRETH3,              # Corresponds to "race_ethnicity"
                     RIDAGEYR,              # Corresponds to "age"
                     INDFMPIR,              # Corresponds to "income"
                     PAD680,                # Corresponds to "sedentary_activity"
                     LBDTCSI,               # Corresponds to "total_cholesterol"
                     depression_score,      # Corresponds to "depression_score"
                     depression_category    # Corresponds to "depression_category"
)
colnames(dat_subset) <- new_colnames
subset_pat<-md.pattern(dat_subset)
vis_miss(dat_raw_2020)
dat_subset_missing_indicators <- dat_subset %>%
  mutate(across(everything(), ~ ifelse(is.na(.), 1, 0), .names = "miss_{col}"))

summary(glm(miss_sleep_trouble ~ BMI, data = dat_subset_missing_indicators, family = binomial
dat_subset_missing_indicators <- dat_subset %>%
  mutate(miss_sleep_trouble = ifelse(is.na(sleep_trouble), 1, 0))

covariate_names <- setdiff(names(dat_subset_missing_indicators), c("sleep_time_weekdays", "mi

results_list <- list()

for (covariate in covariate_names) {
```

```r
  # Define the formula
  formula <- as.formula(paste("miss_sleep_trouble ~", covariate))

  # Fit the model and handle warnings and errors
  tryCatch({
    model <- glm(formula, data = dat_subset_missing_indicators, family = binomial, control =
    if (!model$converged) {
      warning(paste("Model did not converge for covariate:", covariate))
    } else {
      summary_model <- summary(model)
      p_value <- summary_model$coefficients[2, 4]

      # Store the results
      results_list[[covariate]] <- list(
        estimate = summary_model$coefficients[2, 1],
        std_error = summary_model$coefficients[2, 2],
        z_value = summary_model$coefficients[2, 3],
        p_value = p_value
      )
    }
  }, error = function(e) {
    message(paste("Model failed for covariate:", covariate, "with error:", e$message))
  }, warning = function(w) {
    message(paste("Warning for covariate:", covariate, "-", w$message))
  })
}

if (length(results_list) > 0) {
  results_df <- do.call(rbind, lapply(names(results_list), function(name) {
    c(covariate = name, results_list[[name]])
  }))

  print(results_df)
} else {
  message("No models converged successfully.")
}
all_covariates_formula <- as.formula(paste("miss_sleep_trouble ~", paste(covariate_names, co
multivariable_model <- glm(all_covariates_formula, data = dat_subset_missing_indicators, fam
summary(multivariable_model)

imputed_data <- mice(
  dat_subset,
```

```
  m = 5,
  maxit = 10,
  method = 'pmm',
  delta = 0
)
completed_data_2020 <- complete(imputed_data, action = 1)
```

```
colnames(completed_data_2020) <- new_colnames
completed_data_2020[completed_data_2020 == ""] <- NA
completed_data_2020 <- completed_data_2020[,c("sleep_trouble","BMI","avg_MAP","total_choleste
completed_data_2020 <- completed_data_2020 |>
  drop_na() |>
   filter(alcohol < 16 & sleep_trouble != 9) |>
  mutate(across(c(sleep_trouble,gender,race_ethnicity),as.factor)) |>
  mutate(across(c(,),as.numeric))

completed_data_2020$sleep_trouble <- ifelse(completed_data_2020$sleep_trouble == "2",0,1)
imputated_model <- update(final_model_pri, data = completed_data_2020)
summary(imputated_model)
summary(final_model_pri)
```