# Model 01

2024-06-17

#Importing the necessary libraries

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyr)
library(prettyR)
library(dplyr)
library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.3

## Loading required package: lattice

library(rpart)
library(partykit)

## Warning: package 'partykit' was built under R version 4.3.3

## Loading required package: grid

## Loading required package: libcoin

## Warning: package 'libcoin' was built under R version 4.3.3

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.3.3

library(prettyR)
library(dplyr)
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.3.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

## Loading the data file from Wave 2 interviews to calculate the BMI

```
load("34921-0001-Data.rda")

da34921.0001 <- da34921.0001 %>%
  mutate(BMI = ((WEIGHT)/(HEIGHT*HEIGHT) * 703),
         OBESITY = case_when(
           ((WEIGHT)/(HEIGHT*HEIGHT) * 703) >= 30.000 ~ 1,
           ((WEIGHT)/(HEIGHT*HEIGHT) * 703) < 30.000 ~ 0
         ))

obesity <- da34921.0001 %>% select(ID, OBESITY, BMI)
head(obesity)

##       ID OBESITY      BMI
## 1 100005       0 29.63854
## 2 100033       1 33.77728
## 3 100067       0 28.16389
## 4 100080       1 71.40351
## 5 100149       1 38.86545
## 6 100154       0 26.17371
```

## Loading and Processing the Independent Social Network Variables to calculate Bridge from WAVE 1.

```
load("20541-0001-Data.rda")
load("20541-0004-Data.rda")
```

```
da20541.0001 <- da20541.0001 %>%
  select (ID, HEARN_RECODE, GENDER, AGE, RACE_RECODE, ETHGRP, COMBUILD,
DEGREE_RECODE, HISPANIC, MARITLST,JOBSTAT_1, PHYSHLTH, MNTLHLTH, ATNDSERV )

da20541.0001 <- da20541.0001 %>%
  mutate(DEGREE_RECODE = if_else(DEGREE_RECODE == "(-2) don't know", NA,
DEGREE_RECODE),
         HEARN_RECODE = if_else(HEARN_RECODE == "(-2) don't know", NA,
HEARN_RECODE),
         RACE_RECODE = if_else(RACE_RECODE == "(-2) don't know", NA,
RACE_RECODE))

head(da20541.0001)

##       ID       HEARN_RECODE     GENDER AGE         RACE_RECODE
## 1 100005 (4) 100k or higher (2) female  62 (1) white/caucasian
## 2 100033  (2) 25,000-49,999 (2) female  79 (1) white/caucasian
## 3 100080  (3) 50,000-99,999   (1) male  60 (1) white/caucasian
## 4 100154  (2) 25,000-49,999 (2) female  78 (1) white/caucasian
## 5 100203               <NA> (2) female  61 (1) white/caucasian
## 6 100359  (3) 50,000-99,999   (1) male  75 (1) white/caucasian
##                     ETHGRP          COMBUILD
DEGREE_RECODE
## 1               (1) white       (3) average                         (5)
masters
## 2               (1) white (4) above average (2) high school
diploma/equivalency
## 3               (1) white       (3) average (2) high school
diploma/equivalency
## 4               (1) white       (3) average (2) high school
diploma/equivalency
## 5 (3) hispanic, non-black       (3) average                            (1)
none
## 6               (1) white       (3) average (2) high school
diploma/equivalency
##   HISPANIC    MARITLST JOBSTAT_1       PHYSHLTH      MNTLHLTH
## 1   (0) no (1) married   (1) yes (4) very good (4) very good
## 2   (0) no (5) widowed    (0) no (4) very good (4) very good
## 3   (0) no (1) married   (1) yes      (3) good (5) excellent
## 4   (0) no (1) married    (0) no      (3) good      (3) good
## 5  (1) yes (5) widowed   (1) yes      (1) poor      (2) fair
## 6   (0) no (1) married    (0) no      (2) fair      (3) good
##                     ATNDSERV
## 1  (3) several times a year
## 2 (1) less than once a year
## 3            (5) every week
## 4  (6) several times a week
## 5                 (0) never
## 6  (6) several times a week

nrow(da20541.0001)
```

```
## [1] 3005
```

```
da20541.0004 <- da20541.0004 %>%
  group_by(ID) %>%
  filter(n() > 2) %>%
  ungroup()

da20541.0004 <- da20541.0004 %>%
  pivot_longer(
    cols = starts_with("TALKFREQ"),
    names_to = "TALKFREQ",
    values_to = "FREQ"
  )

da20541.0004 <- da20541.0004 %>%
  group_by(ID) %>%
  summarize(
    BRIDGE = if_else(any(FREQ == '(0) have never spoken to each other', na.rm
= TRUE), 1, 0),
    HEALTHDISCUSSIONS = if_else(any(HEALTHTALK == '(3) very likely', na.rm =
TRUE), 1, 0),
    LIVEALONE = if_else(any(LIVEWITH == '(1) yes -- lives in the same
household', na.rm = TRUE), 0,1))

head(da20541.0004)
```

```
## # A tibble: 6 × 4
##   ID     BRIDGE HEALTHDISCUSSIONS LIVEALONE
##   <fct>   <dbl>             <dbl>     <dbl>
## 1 100005      1                 1         0
## 2 100033      0                 1         0
## 3 100080      1                 1         0
## 4 100154      1                 1         0
## 5 100203      0                 1         0
## 6 100359      0                 1         0
```

```
nrow(da20541.0004)
```

```
## [1] 2522
```

```
modeldata <- da20541.0001 %>%
  left_join(da20541.0004, by = "ID")

modeldata <- modeldata %>%
  left_join(obesity, by = "ID")

modeldata<- na.omit(modeldata)
modeldata <- modeldata %>% select(-ID)

modeldata$BRIDGE <- as.factor(modeldata$BRIDGE)
modeldata$HEALTHDISCUSSIONS <- as.factor(modeldata$HEALTHDISCUSSIONS)
```

```
modeldata$LIVEALONE <- as.factor(modeldata$LIVEALONE)
modeldata$OBESITY <- as.factor(modeldata$OBESITY)
head(modeldata)
```

```
##          HEARN_RECODE     GENDER AGE         RACE_RECODE    ETHGRP
## 1 (4) 100k or higher (2) female  62 (1) white/caucasian (1) white
## 2  (2) 25,000-49,999 (2) female  79 (1) white/caucasian (1) white
## 3  (3) 50,000-99,999   (1) male  60 (1) white/caucasian (1) white
## 4  (2) 25,000-49,999 (2) female  78 (1) white/caucasian (1) white
## 7  (2) 25,000-49,999   (1) male  80 (1) white/caucasian (1) white
## 9  (3) 50,000-99,999 (2) female  59 (1) white/caucasian (1) white
##             COMBUILD                               DEGREE_RECODE HISPANIC
MARITLST
## 1       (3) average                               (5) masters   (0) no (1)
married
## 2 (4) above average (2) high school diploma/equivalency   (0) no (5)
widowed
## 3       (3) average (2) high school diploma/equivalency   (0) no (1)
married
## 4       (3) average (2) high school diploma/equivalency   (0) no (1)
married
## 7 (4) above average (2) high school diploma/equivalency   (0) no (5)
widowed
## 9       (3) average (2) high school diploma/equivalency   (0) no (1)
married
##   JOBSTAT_1      PHYSHLTH      MNTLHLTH                       ATNDSERV
BRIDGE
## 1   (1) yes (4) very good (4) very good       (3) several times a year
1
## 2    (0) no (4) very good (4) very good      (1) less than once a year
0
## 3   (1) yes      (3) good (5) excellent                 (5) every week
1
## 4    (0) no      (3) good      (3) good       (6) several times a week
1
## 7    (0) no      (3) good      (3) good                 (5) every week
0
## 9   (1) yes (4) very good (4) very good (2) about once or twice a year
1
##   HEALTHDISCUSSIONS LIVEALONE OBESITY      BMI
## 1                 1         0       0 29.63854
## 2                 1         0       1 33.77728
## 3                 1         0       1 71.40351
## 4                 1         0       0 26.17371
## 7                 1         1       0 24.82300
## 9                 1         0       0 28.48473
```

## Creating Data Partition for 70% Training Data and 30% Testing Data

```
library(rpart)
library(caret)

set.seed(19032023)

index <- createDataPartition(modeldata$OBESITY,
                             p=0.7,
                             list=FALSE,
                             times = 1
                          )

modeldata.train <- modeldata[index,]
modeldata.test <- modeldata[-index,]

nrow(modeldata.train)

## [1] 995

nrow(modeldata.test)

## [1] 425
```

## Applying Logistic Regression on to find the association between Bridge and Obesity.

```
model.lr <- glm(OBESITY ~ BRIDGE, data = modeldata.train, family =
"binomial")

summary.lr <- summary(model.lr)
```

## p-value for Bridge variable

```
print(summary.lr)

##
## Call:
## glm(formula = OBESITY ~ BRIDGE, family = "binomial", data =
modeldata.train)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.30986    0.08817  -3.514 0.000441 ***
## BRIDGE1     -0.33559    0.13131  -2.556 0.010598 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1327.1  on 994  degrees of freedom
## Residual deviance: 1320.5  on 993  degrees of freedom
## AIC: 1324.5
##
## Number of Fisher Scoring iterations: 4
```

## Odds Ratio nnd 95% Confidence Interval

```
odds_ratio <- exp(coef(model.lr)["(Intercept)"])
print(odds_ratio)
```

```
## (Intercept)
##   0.7335526
```

```
conf_int <- exp(confint(model.lr, "(Intercept)"))
```

```
## Waiting for profiling to be done...
```

```
print(conf_int)
```

```
##     2.5 %    97.5 %
## 0.6165970 0.8713551
```

```
predicted.prob.lr <- predict(model.lr, modeldata.test, type = "response")
predicted.obesity.lr <- ifelse(predicted.prob.lr > 0.5, 1, 0)

actual.obesity.lr <- modeldata.test$OBESITY
conf.matrix.lr <- table(Predicted = predicted.obesity.lr, Actual =
actual.obesity.lr)

print(conf.matrix.lr)
```

```
##          Actual
## Predicted   0   1
##         0 261 164
```

```
confusionMatrix(factor(predicted.obesity.lr), factor(modeldata.test$OBESITY),
positive = as.character(1))
```

```
## Warning in confusionMatrix.default(factor(predicted.obesity.lr),
## factor(modeldata.test$OBESITY), : Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 261 164
##          1   0   0
##
```

```
##                Accuracy : 0.6141
##                  95% CI : (0.566, 0.6606)
##     No Information Rate : 0.6141
##     P-Value [Acc > NIR] : 0.5214
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.6141
##              Prevalence : 0.3859
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 1
##
```