# Project 3: Assessing Transportibility of a Cardiovascular Risk Model on the Simulation Data Using NHANES Summary Statistics

Mavis (Xinwen) Liang

## 1 Abstract

This study evaluates the transportability of a cardiovascular risk model developed using Framingham data and applied to NHANES data and its simulated datasets under three different settings. Through statistical analyses, we used Estimated Brier Scores to investigates the model's adaptability to different population datasets. While we found that the NHANES population is slightly healthier than the Framingham population, the transportibility between the two populations appears well. The transportibilies in the simulations with the summary statistics of the NHANES data with or without prior knowledge informed by the Framingham data also performs well. The study highlight the the robustness of the transpotibility measurement in new target populations using only summary data, which can provide analytical results for researchers and governments without assessing the real data.

## 2 Introduction

The primary interest in predictive modeling often lies in its application to specific target populations. Consider, for instance, a healthcare system aiming to utilize a risk prediction model for identifying patients at elevated risk of cardiovascular incidents within its care network. The model is built using the source population and the model to be applied to the target population. A pertinent example is the Framingham ATP-III model (Rubio, Moreno, and Cabrerizo 2004), widely employed to forecast the 10-year risk of cardiovascular events. In this context, a thorough understanding and application of the source model eligibility criteria to the target cohort is an essential component of the research.

In light of these challenges, recent years have seen the emergence of various methodologies aimed at gauging the performance of prediction models within a designated target population. Steingrimsson et al. (2022) has introduce how to assess the transportibility when the outcome

variable is missing in the target population, by estimating the mean squared prediction error. The estimated mean squared error is calculated with the combination of the source data and target data.

However, in practice, we sometimes face the problem of not getting access to the individual data; Instead, we only have the summary data. In this case, if we still want to use our model to predict the outcome of a specific individual, we still have to assess if the model is applicable to the target population. But without the individual data, using only the summary statistics to create a psudo-individual dataset to assess the transportibility, if this transportibility is reliable raises a new question.

In our analysis, we first built a risk score model developed from the Framingham Heart Study data, then we assess whether the transportability of this model to the simulated population based on the NAHNES (National Health and Nutrition Examination Survey) data, attains the similar transportibility as to the true population of NAHNES.

# 3 DATA

## 3.1 Framingham data

The Framingham data obtained from the riskCommunicator package is is a long term prospective study of the etiology of cardiovascular disease among a population of free living subjects in the community of Framingham, Massachusetts (Grembi and Rogawski McQuade 2022). The whole data contains 11627 rows and 39 variables of the participants which have been follow-up for 24 years, with the clinic examinations including cardiovascular disease risk factors and markers of disease such as blood pressure, blood chemistry, lung function, smoking history, health behaviors, ECG tracings, Echocardiography, and medication use.

In order to build the model mimicking a sex-specific multivariable risk factor algorithm built by D'Agostino et al. (2008), we removed the participants that does not have a cardiovascular disease within the 15 years, to see the risk of CVD within the 15th year later since the baseline examination. We also include the variables that is used in the model in the D'Agostino et al. (2008) paper.

The ultimate dataset contain 2539 individuals and 10 variables, which are (1) CVD - whether a participant has a cardiovascular disease at 15th year later since the baseline examination (2) High Density Lipoprotein Cholesterol (mg/dL). (3) TOTCHOL - Serum Total Cholesterol (mg/dL) (4) Age - Age at exam (years), all participants have an age larger than 30 to be accepted in the Framingham study (5) BPMEDS - use of Anti-hypertensive medication at examination (6) SYSBP_UT - Systolic Blood Pressure (mean of last two of three measurements) (mmHg) of participants without the use of Anti-hypertensive medication. (for those with the use of Anti-hypertensive medication, SYSBP_UT is set to 0) (7) SYSBP_T - Systolic Blood Pressure (mean of last two of three measurements) (mmHg) of participants with the use of

Anti-hypertensive medication. (For those without the use of Anti-hypertensive medication, SYSBP_T is set to 0). (8) CURSMOKE - Current cigarette smoking at exam. (9) DIABETES - Diabetic according to criteria of first exam treated or first exam with casual glucose of 200 mg/dL or more. (10) SEX.

### 3.2 NHANES data

In this study, we'll use the NHANES (National Health and Nutrition Examination Survey) data as our target population. NHANES is a program of studies designed to assess the health and nutritional status of adults and children in the United States, which data set is renowned for its comprehensive health and nutritional information collected from a diverse cross-section of the U.S. population. The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions. The examination component consists of medical, dental, and physiological measurements, as well as laboratory tests administered by highly trained medical personnel. The decision to use NHANES data, with its representation of a multi-ethnic population, addressed a critical aspect of model transportability – the evaluation of a prediction model's performance across diverse population groups

To align with the eligibility criteria of the Framingham model, we selected the same variables as in the Framingham dataset and processed them in the same way as the Framingham variables. As in the Framingham study, only participant aging 32 or more will be included, we include a crucial step in the data preparation process involved filtering the NHANES dataset based on age, which enables a more accurate and meaningful comparison between the two datasets. We also remove the missing values in this data in concordance with the source population, where the missingness is relatively small (~30%).

The final NHANES dataset contains 3388 individuals and 9 variables. Outcome (CVD) is unavailable in this data.

## 4 Method

### 4.1 Model building

The model is built with complete cases in Framingham data, given the exploratory nature of this study and its focus on model transportability rather than the development of new predictive algorithms, and also the proportion of missing data was relatively small.

The Framingham data is randomly split into 1:1 training set and test set. The training data is then stratified into two distinct groups based on sex. For each gender subgroup, a logistic regression model was fitted to predict the occurrence of cardiovascular disease (CVD). In the models, continuous covariates are logarithmically transformed to account for the left-skewed,

flat right tail data. Log odds of CVD is model as a linear function of the addicted effect of the continous variables and discrete variables specified above.

Brier score of the model to the training set itself is calculated as a reference for the transportibility analysis.

## 4.2 Transportibility analysis for NHANES individual data

We only use complete cases in NHANES for comparability and consistency with the Framinging data that we used to build the risk model. All the NHANES data is combined with the Framinging's test set to form an ultimate test set to evaluate the transportibility.

We then calculate the estimated brier score for the risk model in the NHANES data. This is done by adjusting the Brier score calculation based on the probability of each observation belonging to the source population, as estimated by a logistic regression model, so that the differences between the source and target populations is taken into account (Steingrimsson et al. 2022). The calculation involves the following steps: (1) *Estimation of Source Membership Probability:* A logistic regression model is fitted to predict the probability of each observation in the combined dataset belonging to the source population. The model uses the same set of predictors as the primary predictive model (log-transformed continuous variables and categorical variables). (2) *Calculation of Inverse Odds:* Inverse odds for each entry of the data $\hat{o}(X_i)$ is derived from the fitted probabilities $\frac{1-p}{p}$ of belonging to the source data. (3) *Model Prediction:* Predictions for the probability of CVD are made for the source dataset using the primary predictive model. (4) *Brier Score Computation:* The estimated brier score for the NHANES data is calculated using the squared differences between the predicted probabilities (of CVD) and the actual outcomes of the Framingham data in the test set. These differences are weighted by the inverse odds, focusing the calculation on how well the model predicts the source population's outcomes. The final score is normalized by the number of observations in the target popupations.

The estimated Brier score we obtain represents a transportability-adjusted measure of the model's prediction accuracy in the source population.

## 4.3 Simulation analysis

We use the ADEMP framework (Morris, White, and Crowther 2019) to explain the simulation:

**Aims:** To evaluate the difference of the transportibility brier scores between the simulating NHANES data and actual NHANES data, we generate individual datasets with the summary statistics provided by the NHANES data, with different underlying assumptions: (1) The

covariates are independently and log-normal distributed; (2) The covariates are normally distributed, and the correlations are informed by the Framingham data; (3) The covariates are log-normally distributed, and the correlations are informed by the Framingham data.

**Data-generating mechanisms:** The NHANES dataset's summary statistics (Table 1), including the mean and standard deviation for continuous variables and percentages for categorical variables, serve as the foundation for our data generation. For each gender group ('MEN' and 'WOMEN'), variables of HDLC, TOTCHOL, AGE, SYSBP_UT, and SYSBP_T are randomly generated, following log-normal distributions or normal distributions with or without correlation (explained in the 3 scenarios below). Categorical variables such as BPMEDS, CURSMOKE, and DIABETES are generated using their respective proportions in the NHANES data. We note that The SYSBP_UT and SYSBP_T variables are conditionally adjusted based on the BPMEDS status, in concordance with how we processes the variables to fit the risk model. We use the sample sizes of NHANES data as the size of data we simulate.

We generate the continous variables in three different settings:

(1) All the continous variables are independently (except for SYSBP_UT and SYSBP_T) and log-normally distributed with parameters $\mu$ and $\sigma$). Parameters $\mu$ and $\sigma$ are derived from the sample mean and sample standard deviation of the NHANES data($\mu = ln\left(\frac{\mu_X^2}{\sqrt{\mu_X^2+\sigma_X^2}}\right)$, $\sigma^2 = ln\left(1+\frac{\sigma_X^2}{\mu_X^2}\right)$, where $\mu_X^2$ and $\sigma_X^2$ is the sample mean and sample variance).

(2) The continuous variables are normally ditributed with the parameters $\mu$ and $\sigma$ equals to the sample means and sample deviation of the NHANES data. The correlation among the variables are informed by the Framingham data. (We use the covariance matrix of the Framingham data).

(3) The continuous variables are log-normally distributed with the parameters calculated the same ways as (1). The correlation of the log-normal variables are informed by the Framingham data. (We use the covariance matrix of the log-transformed continuous variables in the Framingham data.)

We simulate each data 50 times respectively to account for simulation variance.

**Estimands:** We estimate the brier scores of the three settings, which represents a transportability-adjusted measure of the model's prediction accuracy in the simulated target population.

**Methods:** The transportability Brier score of the simulated dataset is calculated in the same way we calculate the Brier score for the actual NHANES data.

**Performance measures:** The efficacy of the simulation under various settings is gauged by comparing the simulated Brier scores to the 'true' Brier score, which is calculated using individual-level NHANES data. We average the brier scores accross different simulations rounds for the three settings respectively to get the final brier scores and the standard deviations. This comparison helps determine the robustness of our model to the data without individual data.

# 5 Results

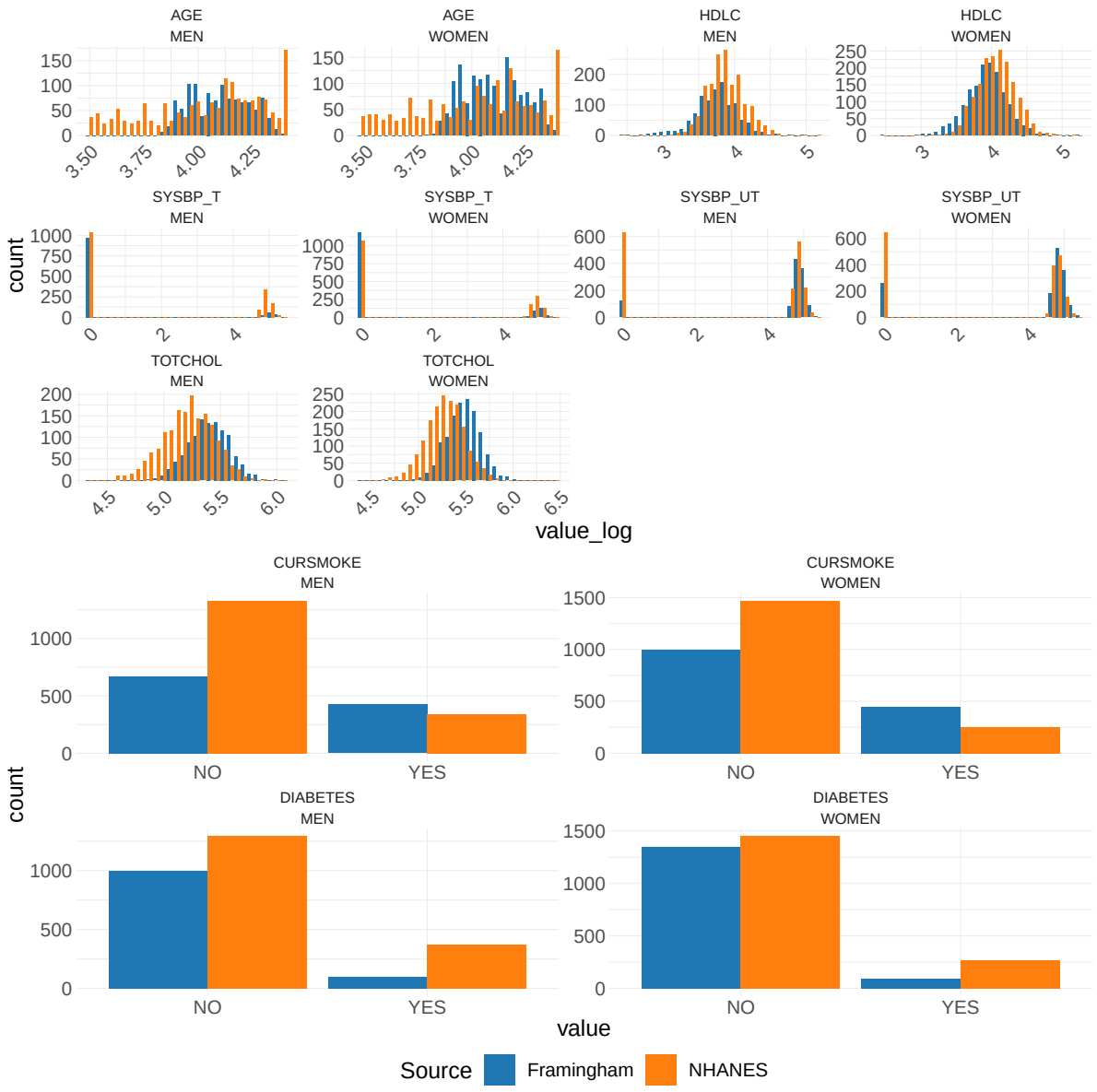


Figure 1: Distribution of the variables in the Framingham population and NHANES population, stratified by sex. The coninous variables are log-transformed.

Figure 1 in the report illustrates the distribution of variables in the Framingham and NHANES populations, stratified by sex. The continuous variables are log-transformed for this analysis. The HDLC, SYSBP_T, SYSBP_UT and TOTCHOL variables appear to be log-normally

Table 1: Summary statistics for NHANES data

| | MEN | WOMEN |
|---|---|---|
| n | 1667 | 1721 |
| HDLC (mean (SD)) | 48.09 (13.92) | 58.76 (16.26) |
| TOTCHOL (mean (SD)) | 185.58 (42.59) | 196.34 (40.97) |
| AGE (mean (SD)) | 57.62 (14.32) | 56.37 (14.40) |
| SYSBP_UT (mean (SD)) | 126.05 (16.60) | 122.06 (18.49) |
| SYSBP_T (mean (SD)) | 134.28 (18.05) | 138.52 (20.77) |
| CURSMOKE = YES (%) | 338 (20.3) | 253 ( 14.7) |
| DIABETES = YES (%) | 373 (22.4) | 269 ( 15.6) |
| BPMEDS = YES (%) | 631 (37.9) | 647 ( 37.6) |

distributed in both populations, which justify our model building. However, the AGE variable appears to be a uniform distribution or a normal distribution with high variance.

It's noted that the NHANES population generally appears healthier than the Framingham population, as indicated by a wider age range, higher HDLCs, smaller Total Cholesterols, higher proportions of non-smokers and higher-proportion of non-diabetes. These is reasonable since in, the Framingham data we used, most participants have an age larger than 47, except for a few that has encounter CVD before the 15th years since the Framingham study begins.

Table 1 provides summary statistics for the NHANES data, detailing mean and standard deviation for continous variables and percentages for categorical variables such as CURSMOKE, DIABETES, and BPMEDS. The mean and standard deviations of SYSBP_UT are calculated with only those that have a BPMEDS = 0, and then mean and standar deviations of SYSBP_T are calculated with only those that have a BPMEDS = 1. The summary statistics in Table 1 is then used to generate the simulation datasets.

Table 2: Estimated Brier Scores

| | MEN | WOMEN |
|---|---|---|
| Framingham | 0.1830495 | 0.1204180 |
| NHANES | 0.11538367 | 0.05767093 |
| sim1..iid.log.normal. | 0.136(0.003) | 0.063(0.003) |
| sim2..normal.with.cov. | 0.145(0.004) | 0.064(0.003) |
| sim3..log.normal.with.cov. | 0.16(0.005) | 0.061(0.002) |

Table 2 shows the Brier scores for different scenarios: the original Framingham model, the NHANES data, and three simulation settings. These scores offer insights into the predictive accuracy of the cardiovascular risk model across different datasets and simulation assumptions.

We see that the Brier score for the NHANES data is very low, indicating that the transportibility of the model from Framingham to NHANES is promising. The Brier scores for both men and women in the NHANES data shrinks as compared to the ones for the Framingham itself, this might be due to the fact that NHANES population is healthier and have way less probabilities of getting CVD.

Comparing the models in different gender, the Brier scores for men is slightly higher in the training set itself, indicating that the model is more accurate when predicting CVD in women. These difference enlarged when applying to the NHANES population (the true population and simulated population), suggesting that the model can much better in predivting CVD of women.

Comparing different simulation settings, we found that the Brier scores are slightly higher than the actual NHANES data, but they are roughly similar, even we simly assume that the data are independently distributed. The variation in different simulation rounds is small, suggesting that the Brier score estimates are robust with the same summary statistics. When we add more information that is provided by the Framingham data, the brier scores get closer to the one for Framingham data.(sim3 vs sim1 and sim3 vs sim2).

The results combined suggest an overall good transportibility from the Framingham risk model to the NHANES population. And the transportibility is robust accross different simulation settings using summary statistics from the NHANES data, and as compared to the original data.

# 6 Discussion

**Model Transportability and Adaptability:** The study's findings highlight the complexities of model transportability and adaptation to different population datasets. The differences in Brier scores between the Framingham and NHANES data suggest a variation in model performance across populations.

**Impact of Distribution Assumptions:** The simulation results demonstrate how different assumptions about variable distributions (independent, normal with covariance, log-normal with covariance) affect the model's predictive performance. This underlines the importance of considering underlying data distribution characteristics when generalizing models to new populations.

**Methodological Considerations:** The use of log-transformed variables and handling of categorical variables like BPMEDS, CURSMOKE, and DIABETES provide methodological insights into dealing with diverse data types in predictive modeling.

**Limitations and Future Work:** To further investigate if we can assess transportibility without real data, using only summary statistics of the data, we should do the same analysis with different datasets besides NHANES, preferably with a true outcome variable.

# 7 Reference


D'Agostino, Ralph, Ramachandran Vasan, Michael Pencina, Philip Wolf, Mark Cobain, Joseph Massaro, and William Kannel. 2008. "General Cardiovascular Risk Profile for Use in Primary Care: The Framingham Heart Study." *Circulation* 117 (March): 743–53. https://doi.org/10.1161/CIRCULATIONAHA.107.699579.

Grembi, Jess, and Elizabeth Rogawski McQuade. 2022. "Introducing riskCommunicator: An r Package to Obtain Interpretable Effect Estimates for Public Health." *PLOS ONE* 17 (July): e0265368. https://doi.org/10.1371/journal.pone.0265368.

Morris, Tim, Ian White, and Michael Crowther. 2019. "Using Simulation Studies to Evaluate Statistical Methods." *Statistics in Medicine* 38 (January). https://doi.org/10.1002/sim.8086.

Rubio, Miguel, C. Moreno, and L. Cabrerizo. 2004. "Guidelines for Dyslipemias Treatment: Adult Treatment Panel III (ATP-III)." *Endocrinologia y Nutricion* 51 (May): 254–65.

Steingrimsson, Jon A, Constantine Gatsonis, Bing Li, and Issa J Dahabreh. 2022. "Transporting a Prediction Model for Use in a New Target Population." *American Journal of Epidemiology* 192 (2): 296–304. https://doi.org/10.1093/aje/kwac128.

# 8 Code Appendix

```
library(riskCommunicator)
library(nhanesA)
library(MASS)
library(compositions)
library(tidyverse)
library(tableone)
library(kableExtra)
library(gridExtra)
#source("C:/Users/xliang34/OneDrive - Brown University/2023/missing_heatmap_function.R")
knitr::opts_chunk$set(warning = FALSE, message = FALSE,
                      echo = FALSE, fig.align = "center")
# Pre-processing
data("framingham")
# The Framingham data has been used to create models for cardiovascular risk.
# The variable selection and model below are designed to mimic the models used
# in the paper General Cardiovascular Risk Profile for Use in Primary Care
# This paper is available (cvd_risk_profile.pdf) on Canvas.

framingham_df <- framingham %>% dplyr::select(c(CVD, TIMECVD, SEX, TOTCHOL, AGE,
                                    SYSBP, DIABP, CURSMOKE, DIABETES, BPMEDS,
                                    HDLC, BMI))
framingham_df <- na.omit(framingham_df)

#CreateTableOne(data=framingham_df, strata = c("SEX"))

# Get blood pressure based on whether or not on BPMEDS
framingham_df$SYSBP_UT <- ifelse(framingham_df$BPMEDS == 0,
                                 framingham_df$SYSBP, 0)
framingham_df$SYSBP_T <- ifelse(framingham_df$BPMEDS == 1,
                                framingham_df$SYSBP, 0)

# Looking at risk within 15 years - remove censored data
#dim(framingham_df)
framingham_df <- framingham_df %>%
  filter(!(CVD == 0 & TIMECVD <= 365*15)) %>%
  dplyr::select(-c(TIMECVD))

# The NHANES data here finds the same covariates among this national survey data
# blood pressure, demographic, bmi, smoking, and hypertension info
```

```r
bpx_2017 <- nhanes("BPX_J") %>%
  dplyr::select(SEQN, BPXSY1 ) %>%
  rename(SYSBP = BPXSY1)
demo_2017 <- nhanes("DEMO_J") %>%
  dplyr::select(SEQN, RIAGENDR, RIDAGEYR) %>%
  rename(SEX = RIAGENDR, AGE = RIDAGEYR)
bmx_2017 <- nhanes("BMX_J") %>%
  dplyr::select(SEQN, BMXBMI) %>%
  rename(BMI = BMXBMI)
smq_2017 <- nhanes("SMQ_J") %>%
  mutate(CURSMOKE = case_when(SMQ040 %in% c(1,2) ~ 1,
                              SMQ040 == 3 ~ 0,
                              SMQ020 == 2 ~ 0)) %>%
  dplyr::select(SEQN, CURSMOKE)
bpq_2017 <- nhanes("BPQ_J") %>%
  mutate(BPMEDS = case_when(
    BPQ020 == 2 ~ 0,
    BPQ040A == 2 ~ 0,
    BPQ050A == 1 ~ 1,
    TRUE ~ NA )) %>%
  dplyr::select(SEQN, BPMEDS)
tchol_2017 <- nhanes("TCHOL_J") %>%
  dplyr::select(SEQN, LBXTC) %>%
  rename(TOTCHOL = LBXTC)
hdl_2017 <- nhanes("HDL_J") %>%
  dplyr::select(SEQN, LBDHDD) %>%
  rename(HDLC = LBDHDD)
diq_2017 <- nhanes("DIQ_J") %>%
  mutate(DIABETES = case_when(DIQ010 == 1 ~ 1,
                              DIQ010 %in% c(2,3) ~ 0,
                              TRUE ~ NA)) %>%
  dplyr::select(SEQN, DIABETES)

# Join data from different tables
df_2017 <- bpx_2017 %>%
  full_join(demo_2017, by = "SEQN") %>%
  full_join(bmx_2017, by = "SEQN") %>%
  full_join(hdl_2017, by = "SEQN") %>%
  full_join(smq_2017, by = "SEQN") %>%
  full_join(bpq_2017, by = "SEQN") %>%
  full_join(tchol_2017, by = "SEQN") %>%
```

```
  full_join(diq_2017, by = "SEQN")

df_2017$SYSBP_UT <- ifelse(df_2017$BPMEDS == 0,
                                  df_2017$SYSBP, 0)
df_2017$SYSBP_T <- ifelse(df_2017$BPMEDS == 1,
                                 df_2017$SYSBP, 0)
df_2017_final <- df_2017 %>% dplyr::select("HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
                                    "SYSBP_T", "CURSMOKE", "DIABETES", "SEX",
                                    "BPMEDS")
N_final <- na.omit(df_2017_final) %>%
  mutate(BPMEDS = ifelse(BPMEDS == 0, "NO", "YES"),
         CURSMOKE = ifelse(CURSMOKE == 0, "NO", "YES"),
         DIABETES = ifelse(DIABETES == 0, "NO", "YES"),
         SEX = ifelse(SEX == 1, "MEN", "WOMEN")) %>%
  filter(AGE>=32)


F_final <- framingham_df %>%
  dplyr::select("CVD","HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
         "SYSBP_T", "CURSMOKE", "DIABETES", "SEX") %>%
  mutate(CURSMOKE = ifelse(CURSMOKE == 0, "NO", "YES"),
         DIABETES = ifelse(DIABETES == 0, "NO", "YES"),
         SEX = ifelse(SEX == 1, "MEN", "WOMEN"))
# Model building
# Split the Framingham data into 1:1 training set and test set.
# Only use the trainingset to build the model
set.seed(5)
train_index <-
  sample(1:nrow(F_final),
         size = nrow(F_final)/2,
         replace = F)
F_train <- F_final[train_index, ]
F_test <- F_final[-train_index,]
# Filter to each sex
F_train_men <- F_train %>% filter(SEX == "MEN")
F_train_women <- F_train %>% filter(SEX == "WOMEN")

# Fit models with log transforms for all continuous variables
mod_men <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
      data= F_train_men, family= "binomial")
```

```r
mod_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= F_train_women, family= "binomial")

# Calculate the brier scores in the training set
F_train_men$predicted_prob <- predict(mod_men, type = "response")
F_train_women$predicted_prob <- predict(mod_women, type = "response")

brier_train_men <- mean((F_train_men$CVD - F_train_men$predicted_prob)^2)
brier_train_women <- mean((F_train_women$CVD - F_train_women$predicted_prob)^2)
#' Function for calculating the estimated brier score
#'
#' @param mod The model that built with the training set
#' @param df_test_target The target test set. True CVD is not needed.
#' @param df_test_source The source test set. True CVD is needed.
cal_brier <- function(mod, df_test_target, df_test_source){
  df_test_target <- df_test_target%>% mutate(S=0)
  df_test_source <- df_test_source%>% mutate(S=1)
  df_test_combo <- rbind(df_test_source %>% dplyr::select(-CVD),
                         df_test_target)
  fitted_S <- glm(S~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                    log(SYSBP_T+1)+CURSMOKE+DIABETES,
                    data= df_test_combo, family= "binomial")$fitted.values
  inv_odds <- (1-fitted_S)/fitted_S

  pred_Y <- predict(mod, newdata = df_test_source, type = "response")
  true_Y_test <- df_test_source$CVD

  brier <- sum(inv_odds[df_test_combo$S==1] * (true_Y_test - pred_Y)^2) / sum(df_test_comb
  return(brier)
}
combined_data <- rbind((F_final[, !names(F_final) %in%
                                  c("CVD")]) %>% mutate(Source = 'Framingham'),
                       N_final %>% mutate(Source = 'NHANES') %>% dplyr::select(-BPMEDS))

continuous_vars <- c("HDLC", "TOTCHOL", "AGE", "SYSBP_UT", "SYSBP_T")
discrete_vars <- c("CURSMOKE", "DIABETES")

# For continuous variables
combined_long_continuous <-
```

```r
  combined_data %>%
  gather(key = "variable", value = "value", one_of(continuous_vars)) %>%
  mutate(value_log = log(value + 1))

# For discrete variables
combined_long_discrete <- combined_data %>%
    gather(key = "variable", value = "value", one_of(discrete_vars))

hist_plot <-
ggplot(combined_long_continuous, aes(x = value_log, fill = Source)) +
  geom_histogram(position = "dodge", bins = 30) +
  facet_wrap(~ variable + SEX, scales = "free") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        text = element_text(size = 25),
        axis.title = element_text(size = 30),
        axis.text = element_text(size = 25),
        legend.position = "none") +
  scale_fill_manual(values = c("#1f77b4", "#ff7f0e"))

bar_plot <-
ggplot(combined_long_discrete, aes(x = value, fill = Source)) +
    geom_bar(position = "dodge") +
    facet_wrap(~ variable + SEX, scales = "free") +
    theme_minimal() +
  theme(text = element_text(size = 25),
        axis.title = element_text(size = 30),
        axis.text = element_text(size = 25),
        legend.title = element_text(size = 30),
        legend.text = element_text(size = 25),
        legend.key.size = unit(1.5, "cm"),
        legend.position = "bottom") +
    scale_fill_manual(values = c("#1f77b4", "#ff7f0e"))

grid.arrange(hist_plot, bar_plot)
cto_object <- CreateTableOne(data = N_final, strata = c("SEX"), test = FALSE)
## Adjust the mean and sd of SYSBP_T
new_SYSBP_table <- N_final %>%
  group_by(SEX, BPMEDS) %>%
  summarise(mean_UT = mean(SYSBP_UT), mean_T = mean(SYSBP_T),
            sd_UT = sd(SYSBP_UT), sd_T = sd(SYSBP_T))
```

```r
cto_object$ContTable[["MEN"]]["SYSBP_UT","mean"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="MEN", BPMEDS=="NO"))$m
cto_object$ContTable[["MEN"]]["SYSBP_UT","sd"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="MEN", BPMEDS=="NO"))$s
cto_object$ContTable[["WOMEN"]]["SYSBP_UT","mean"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="WOMEN", BPMEDS=="NO"))
cto_object$ContTable[["WOMEN"]]["SYSBP_UT","sd"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="WOMEN", BPMEDS=="NO"))

cto_object$ContTable[["MEN"]]["SYSBP_T","mean"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="MEN", BPMEDS=="YES"))$
cto_object$ContTable[["MEN"]]["SYSBP_T","sd"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="MEN", BPMEDS=="YES"))$
cto_object$ContTable[["WOMEN"]]["SYSBP_T","mean"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="WOMEN", BPMEDS=="YES")
cto_object$ContTable[["WOMEN"]]["SYSBP_T","sd"] <- (new_SYSBP_table %>%
                                                       filter(SEX=="WOMEN", BPMEDS=="YES")

summary_tb <- rbind(print(cto_object$ContTable,
                          noTests = TRUE, printToggle = FALSE) ,
                    print(cto_object$CatTable, noTests = TRUE,
                    printToggle = FALSE))[-c(7,10),]
#Display
summary_tb %>%
  kable(format = "latex", booktabs = TRUE,
        caption = "Summary statistics for NHANES data") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
# We use all the NHANES data and combine it with the test set from Framingham data to crea
brier_test_men <- cal_brier(mod = mod_men,
          df_test_target = N_final %>% filter(SEX=="MEN") %>% dplyr::select(-BPMEDS),
          df_test_source = F_test %>% filter(SEX=="MEN"))
brier_test_women  <- cal_brier(mod = mod_women,
        df_test_target = N_final %>% filter(SEX=="WOMEN") %>% dplyr::select(-BPMEDS),
        df_test_source = F_test %>% filter(SEX=="WOMEN"))
# Based on the summary statistics from the createTableOne_object of the NHANES data, a dat
sim_function <- function(createTableOne_object, method = "independent"){
  df <- NULL
  for (i in c("MEN", "WOMEN")) {
    mean <- createTableOne_object$ContTable[[i]][,"mean"]
    sd <- createTableOne_object$ContTable[[i]][,"sd"]
```

```
prop_BPMEDS <- createTableOne_object$CatTable[[i]]$BPMEDS[2,"percent"]/100
prop_CURSMOKE <- createTableOne_object$CatTable[[i]]$CURSMOKE[2,"percent"]/100
prop_DIABETES <- createTableOne_object$CatTable[[i]]$DIABETES[2,"percent"]/100
n <- createTableOne_object$ContTable[[i]][1,"n"]

mu <- log(mean^2/sqrt(mean^2 + sd^2))
sigma <- sqrt(log(1 + sd^2 / mean^2))
if(method=="independent"){
  HDLC <- rlnorm(n, mu["HDLC"], sigma["HDLC"])
  TOTCHOL <- rlnorm(n, mu["TOTCHOL"], sigma["TOTCHOL"])
  AGE <- rlnorm(n, mu["AGE"], sigma["AGE"])
  SYSBP_UT = rlnorm(n, mu["SYSBP_UT"], sigma["SYSBP_UT"])
  SYSBP_T = rlnorm(n, mu["SYSBP_T"], sigma["SYSBP_T"])
}else if(method == "cov"){
  cov_F <- cov(log(F_final %>%
                     filter(SEX==i) %>%
                     dplyr::select("HDLC", "TOTCHOL", "AGE",
                                   "SYSBP_UT", "SYSBP_T")
                   +1 ))
  simulated_continous <- as.data.frame <- rlnorm.rplus(n,mu,cov_F)
  HDLC <- simulated_continous[, 1]
  TOTCHOL <- simulated_continous[,2]
  AGE <- simulated_continous[,3]
  SYSBP_UT <- simulated_continous[, 4]
  SYSBP_T <- simulated_continous[, 5]
}else if(method == "normal"){
  cov_F <- cov(F_final %>% filter(SEX==i) %>%
                     dplyr::select("HDLC", "TOTCHOL", "AGE",
                                   "SYSBP_UT", "SYSBP_T"))
  simulated_continous <- as.data.frame <- mvrnorm(n,mean,cov_F)
  HDLC <- simulated_continous[, 1]
  TOTCHOL <- simulated_continous[,2]
  AGE <- simulated_continous[,3]
  SYSBP_UT <- simulated_continous[, 4]
  SYSBP_T <- simulated_continous[, 5]
}


BPMEDS <- sample(c("YES", "NO"), size = n, replace = TRUE,
                 prob = c(prop_BPMEDS, 1 - prop_BPMEDS))
SYSBP_UT = ifelse(BPMEDS=="NO", SYSBP_UT, 0)
```

```
    SYSBP_T = ifelse(BPMEDS=="YES", SYSBP_T, 0)

    CURSMOKE = sample(c("YES", "NO"), size = n, replace = TRUE,
                      prob = c(prop_CURSMOKE, 1 - prop_CURSMOKE))
    DIABETES = sample(c("YES", "NO"), size = n, replace = TRUE,
                      prob = c(prop_DIABETES, 1 - prop_DIABETES))

    df <- rbind(df, data.frame(HDLC, TOTCHOL,AGE, SYSBP_UT,
                      SYSBP_T, CURSMOKE, DIABETES, SEX=i))

  }
  return(df)
}
sim_ntimes <- function(sim, method="independent", n){
  brier_sim_men <- NULL
  brier_sim_women <- NULL
  for (i in 1:n) {
  sim_df <- sim(cto_object, method)
  brier_sim_men <- c(brier_sim_men, cal_brier(mod = mod_men,
          df_test_target = sim_df %>% filter(SEX=="MEN"),
          df_test_source = F_test %>% filter(SEX=="MEN")))
  brier_sim_women  <- c(brier_sim_women,
                        cal_brier(mod = mod_women,
                          df_test_target = sim_df %>% filter(SEX=="WOMEN"),
                          df_test_source = F_test %>% filter(SEX=="WOMEN")))
  }
  return(list(mean_sim_brier_MEN = round(mean(brier_sim_men),3),
              sd_sim_brier_MEN = round(sd(brier_sim_men),3),
              mean_sim_brier_WOMEN = round(mean(brier_sim_women),3),
              sd_sim_brier_WOMEN = round(sd(brier_sim_women),3)))
}

set.seed(6)
sim1 <- sim_ntimes(sim_function, n = 50)
sim2 <- sim_ntimes(sim_function, method = "normal", 50)
sim3 <- sim_ntimes(sim_function, method = "cov", 50)

# Display the results
data.frame(Framingham = c(brier_train_men, brier_train_women),
           NHANES = c(brier_test_men, brier_test_women),
           `sim1 (iid log-normal)` = c(paste0(sim1$mean_sim_brier_MEN, "(",
```

```
                          sim1$sd_sim_brier_MEN, ")"),
                    paste0(sim1$mean_sim_brier_WOMEN, "(",
                           sim1$sd_sim_brier_WOMEN, ")")),
          `sim2 (normal with cov)` = c(paste0(sim2$mean_sim_brier_MEN, "(",
                          sim2$sd_sim_brier_MEN, ")"),
                    paste0(sim2$mean_sim_brier_WOMEN, "(",
                           sim2$sd_sim_brier_WOMEN, ")")),
          `sim3 (log-normal with cov)` = c(paste0(sim3$mean_sim_brier_MEN, "(",
                          sim3$sd_sim_brier_MEN, ")"),
                    paste0(sim3$mean_sim_brier_WOMEN, "(",
                           sim3$sd_sim_brier_WOMEN, ")")),
          row.names = c("MEN", "WOMEN")) %>%
t() %>%
kable(format = "latex", booktabs = TRUE,
      caption = "Estimated Brier Scores") %>%
kable_styling(latex_options = c("striped", "hold_position"))
```