Transporting the framingham risk prediction model for use in a new target population

Kerry Ye

2023-11-30

Abstract

Objective: The source data often can not treated as a random sample from the target population due to differences in covariate distributions between the two populations. Transportation analysis techniques can overcome this challenge. In this project, our goals are 1) to evaluate the Cardiovascular Disease Risk Prediction model's performance on individual-level NHANES data, originally developed from the Framingham Heart Study, and 2) to assess the model's performance on simulated NHANES data when actual data are unavailable. We will consider three simulation scenarios: observed Framingham correlation, no correlation, and high correlation (0.6).

Methods: We employ a logistic regression model, developed from the Framingham dataset, which is trained using the training set. This model is used to predict the risk of cardiovascular disease, with each version of the model stratified by sex. Additionally, we utilize logistic regression to estimate the inverse-odds weighting estimator by calculating $\Pr[S = 1 \mid X, D_{\text{test},i} = 1]$, and assessed the performance of each sex-stratified risk score prediction model using modified Brier Score when the target population is different from the source population.

Results: The Brier scores for the women's risk score prediction model were consistently lower than those for the men's model. Notably, the Brier score for the men's model using simulated NHANES data was higher than that using actual NHANES data. The women's model performs well with no and observed correlation in the simulated data, showing slightly lower Brier scores than the actual NHANES data. The high correlation scenario for the women's model leads to a significantly higher predictive error.

Introduction

This project, in collaboration with Dr. Jon Steingrimsson, aims to investigate the performance of a prediction model in a target population different from the one originally used for model development. Prediction models, typically built from source study data like randomized trials or large observational databases, may not represent a random sample from the target population due to covariate distribution differences.

Recently, transportation analysis methods have been developed to evaluate the performance of transferring a prediction model to a new target population. These techniques are applicable when covariate and outcome data are available in the source population, but only outcome data are unavailable in the target population. They adjust for distribution differences between the source and target populations through weighting estimators, facilitating the use of prediction models in various populations.[3] However, when target population data are unavailable, researchers may rely solely on summary statistics to simulate data. Therefore, comparing the model's performance in actual and simulated target populations is crucial.

This project's objectives are: 1) to evaluate the Cardiovascular Disease Risk Prediction model's performance on individual-level NHANES data, originally developed from the Framingham Heart Study, and 2) to assess the model's performance on simulated NHANES data when actual data are unavailable. We will consider three simulation scenarios: observed Framingham correlation, no correlation, and high correlation (0.6).

Data Source

We compare the Framingham data and the NHANES data in the following sections:

Framingham Dataset

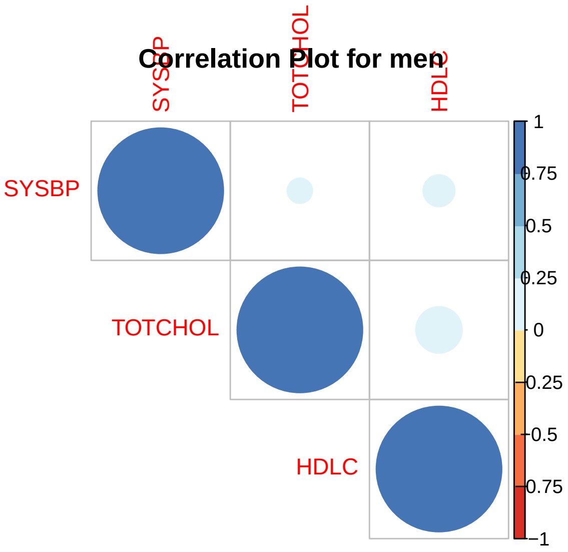
Framingham: The Framingham Heart Study is a long-term investigation of cardiovascular disease among a population in Framingham, Massachusetts. This study was the first of its kind to prospectively examine cardiovascular disease, identifying the concept of risk factors and their combined effects. Initiated in 1948, the study enrolled 5,209 subjects. Participants have been examined biennially, and all subjects are continuously monitored for cardiovascular outcomes. Clinical examination data includes cardiovascular disease risk factors and markers, such as blood pressure, blood chemistry, lung function, smoking history, health behaviors, ECG tracings, echocardiography, and medication use. The enclosed dataset represents a subset of the data collected in the Framingham study, comprising laboratory, clinical, questionnaire, and adjudicated event data[1].

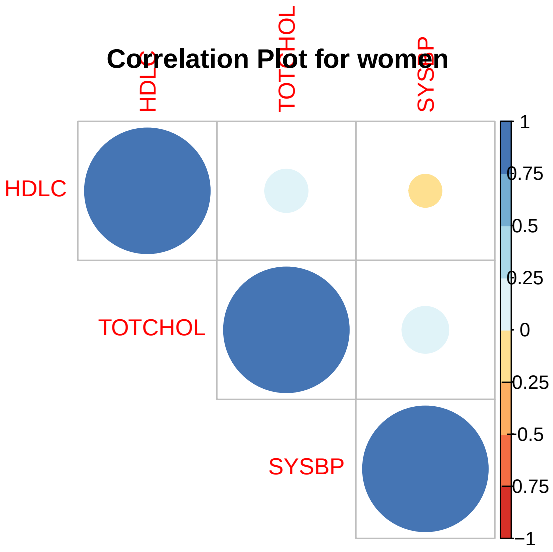
In our study, cardiovascular disease (CVD) is the primary outcome of interest. We excluded individuals with censor data within 15 years leading up to the CVD event. This resulted in a complete dataset of n=2539 participants without missing information, specifically comprising n=1094 males and n=1445 females. Our study incorporated the Framingham study criteria, including Sex (SEX), Serum Total Cholesterol (TOTCHOL), High-Density Lipoprotein Cholesterol (HDLC), age at exam (AGE), use of anti-hypertensive medication (BPMEDS), Systolic Blood Pressure (SYSBP), Diabetes (DIABETES), and current cigarette smoking status (CURSMOKE). SYSBP_UT represents the systolic blood pressure for individuals not using anti-hypertensive medication, while SYSBP_T is for those on such medication. These variables were used as covariates in the CVD prediction model.

	1	2	p-value
n	1094	1445	
CVD (mean (SD))	0.33 (0.47)	0.17 (0.37)	<0.001
SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001
TOTCHOL (mean (SD))	226.44 (41.49)	246.32 (45.51)	<0.001
AGE (mean (SD))	60.01 (8.18)	60.55 (8.40)	0.106
SYSBP (mean (SD))	138.94 (20.89)	139.94 (23.71)	0.272
CURSMOKE (mean (SD))	0.39 (0.49)	0.31 (0.46)	<0.001
DIABETES (mean (SD))	0.09 (0.28)	0.07 (0.25)	0.037
BPMEDS (mean (SD))	0.11 (0.32)	0.18 (0.38)	<0.001
HDLC (mean (SD))	43.63 (13.37)	53.07 (15.67)	<0.001
SYSBP_UT (mean (SD))	121.04 (46.69)	111.49 (55.89)	<0.001
SYSBP_T (mean (SD))	17.90 (50.93)	28.45 (61.53)	<0.001

Table 1: Summary Statistics Stratified by SEX

}





NHANES Dataset

NHANES: The National Health and Nutrition Examination Survey (NHANES) is a national survey conducted to monitor the health and nutritional status of both adults and children in the United States. What sets NHANES apart is its combination of interviews and physical examinations. The survey is administered by a part of the Centers for Disease Control and Prevention (CDC). The first NHANES program was initiated in 1960. Since 1999, the survey has examined about 5,000 individuals yearly in 15 different counties across the country. The data collected by NHANES is instrumental in providing vital health statistics [2].

The original NHANES dataset did not include cardiovascular disease (CVD) outcomes, totaling n=9254 participants, with n=4557 males and n=4697 females. The dataset encompasses variables like SEX, TOTCHOL, HDLC, AGE, BPMEDS, SYSBP, DIABETES, CURSMOKE, SYSBP_UT, and SYSBP_T, similar to the Framingham study. The 'Missingness in NHANES Dataset' table shows the highest missingness in CURSMOKE, BPMEDS, SYSBP, HDLC, and TOTCHOL, ranging from 36.72% to 27.19%. Assuming the missing data pattern is Missing at Random (MAR), we excluded cases with missing values for simplicity and efficiency. Consequently, the complete NHANES dataset was reduced to n=4310 participants, consisting of n=2105 males and n=2205 females.

Table 2: Summary Statistics Stratified by SEX

	1	2	p
n	2105	2205	
SYSBP (mean (SD))	126.44 (16.83)	123.70 (20.36)	<0.001
SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001
AGE (mean (SD))	50.15 (18.83)	48.90 (18.57)	0.029
HDLC (mean (SD))	48.11 (13.59)	58.10 (15.68)	<0.001
CURSMOKE (mean (SD))	0.20 (0.40)	0.14 (0.35)	<0.001
BPMEDS (mean (SD))	0.30 (0.46)	0.29 (0.45)	0.584
TOTCHOL (mean (SD))	183.10 (41.65)	190.51 (41.20)	<0.001
DIABETES (mean (SD))	0.18 (0.38)	0.12 (0.33)	<0.001
SYSBP_UT (mean (SD))	86.46 (57.73)	83.64 (55.42)	0.101
SYSBP_T (mean (SD))	39.98 (62.19)	40.07 (63.62)	0.964

Table 3: Missingness in NHANES Dataset

Variable	Number	Percentage
CURSMOKE	3398	36.72%
BPMEDS	3398	36.72%
SYSBP	2952	31.9%
HDLC	2516	27.19%
TOTCHOL	2516	27.19%
BMI	1249	13.5%
DIABETES	361	3.9%

Methodology

Model

The prediction model presented below is a logistic regression model designed to assess the risk factors for cardiovascular disease (CVD) using data from the Framingham Heart Study. This model is fitted separately for men and women for subsequent analysis:

$$\begin{aligned}\text{logit}(\Pr(\text{CVD} = 1)) = \beta_0 &+ \beta_1 \log(\text{HDLC}) + \beta_2 \log(\text{TOTCHOL}) \\ &+ \beta_3 \log(\text{AGE}) + \beta_4 \log(\text{SYSBP\_UT} + 1) \\ &+ \beta_5 \log(\text{SYSBP\_T} + 1) + \beta_6 \text{CURSMOKE} \\ &+ \beta_7 \text{DIABETES}\end{aligned}$$

The outcome variable of interest is a binary indicator indicating whether cardiovascular disease occurred (CVD=1) or did not occur (CVD=0) during the follow-up period, and the logit function is used to denote the logistic transformation, which is the natural log of the odds of the event (CVD = 1 in this case).

The model includes several continuous variables: High-Density Lipoprote in Cholesterol (HDLC) in mg/dL, Serum Total Cholesterol (TOTCHOL) measured in mg/dL, and age at exam (AGE) in years. Additionally, the model accounts for the use of anti-hypertensive medication (BPMEDS), distinguishing between those who currently use these medications and those who do not. Systolic Blood Pressure (SYSBP) taken as the mean of the last two of three measurements in mmHg. SYSBP is further bifurcated into two variables: SYSBP_UT and SYSBP_T. SYSBP_UT represents the systolic blood pressure for individuals not currently using anti-hypertensive medication, while SYSBP_T represents the systolic blood pressure for individuals who are on such medication. These variables undergo logarithmic transformation to approximate normality, addressing skewness in the data.

There are also two categorical variables: Diabetes (DIABETES), where 0 indicates non-diabetic and 1 indicates diabetic based on the criteria of the first exam treated or a casual glucose level of 200 mg/dL or more; and current cigarette smoking status (CURSMOKE), with 0 representing non-smokers and 1 representing current smokers.

Metrics

As traditional performance metrics are not suitable for evaluating the use of prediction models developed from source data in a target population, Steingrimsson, et al. proposed a modified version of the Brier Score[1].

Let S be an indicator for the population from which data are obtained, with S = 1 for the source population and S = 0 for the target population. Let D_{test} denote the test set indicator, where $D_{\text{test}} = 1$ if an observation is in either the source or target test set. Denote n = n_{source} + n_{target} as the sample size of the composite dataset consisting of data from both the source and target population samples. Use the function $g_{\hat{\beta}}(X)$ to denote the "fitted" model with estimated parameter $\hat{\beta}$.

With the inverse-odds weighting estimator, the Brier Score for the target population is defined as follows:

$$\hat{\psi}_{\hat{\beta}} = \frac{\sum_{i=1}^{n} I(S_i = 1, D_{\text{test},i} = 1)\hat{o}(X_i)(Y_i - g_{\hat{\beta}}(X)))^2}{\sum_{i=1}^{n} I(S_i = 0, D_{\text{test},i} = 1)}$$

,

where the function $\hat{o}(X_i)$ is the inverse-odds weighting estimator in the test set, defined as $\frac{\Pr[S=0|X,D_{\text{test},i}=1]}{\Pr[S=1|X,D_{\text{test},i}=1]}$.

Analysis

We divided the project goal into two analyses. Since we only consider complete cases in this project, we omitted the missing values present in the NHANES data.

‘

Analysis I

The second goal of this project is to evaluate the performance of the risk score model in the NHANES study, which was developed using data from the Framingham Heart Study.

Initially, we created a composite dataset by combining the Framingham and NHANES datasets. It's important to note that the Framingham data includes both outcome and covariate information, whereas the NHANES dataset contains only covariate data. We then stratified the composite dataset into two groups: men and women. For each group, we first stratified the composite dataset into two groups: men and women

Initially, we stratified the Framingham dataset into two groups: men and women. For each group, we partitioned the data into training and testing sets by randomly selecting 70% of the data for training and the remaining 30% for testing. We then created a composite test dataset by combining the Framingham test set and NHANES datasets for each set. The training data was used to fit the prediction model as previously described, and the testing data was used for performance evaluation. Since the NHANES data is not a random sample from the Framingham data, and these two populations can exhibit very different data distributions, we employed the modified version of the Mean Squared Error (MSE) as previously described. This method evaluates the performance of the risk score prediction model developed from the Framingham data when applied to the NHANES study. The inverse odds weighting estimator is estimated using logistic regression to determine $\Pr[S = 1 \mid X, D_{\text{test},i} = 1]$ in the test set. Eventually, we calculate two Brier scores to evaluate the risk score prediction model, stratified by sex.

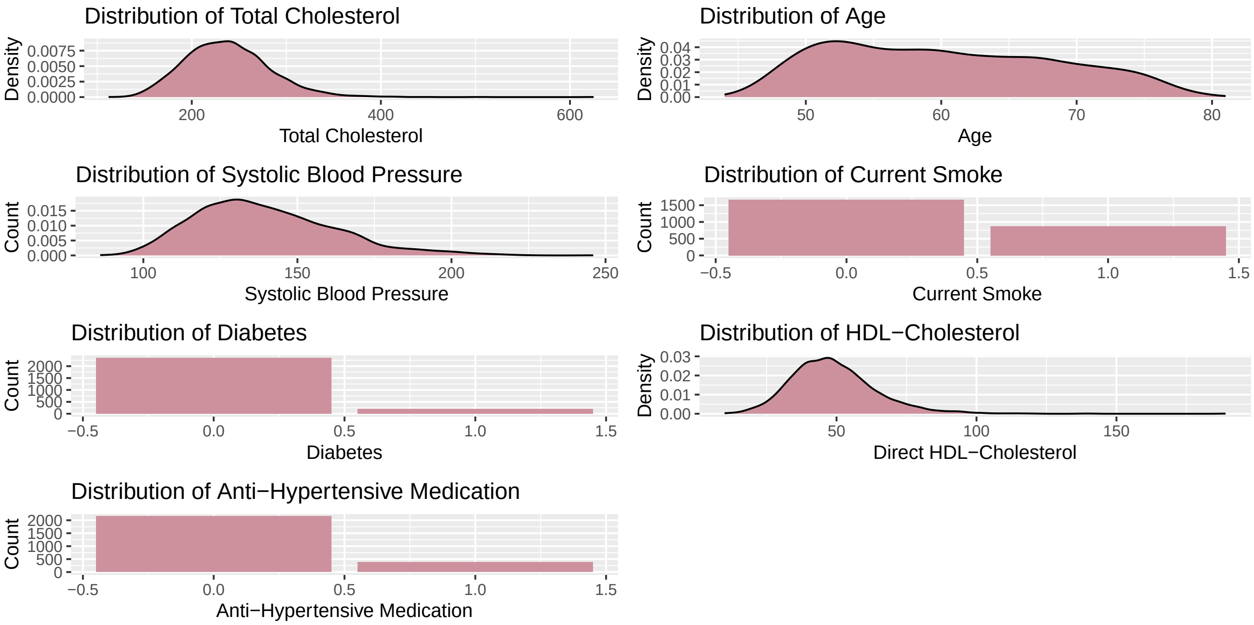
Analysis II

We would discuss the simulation process by using the following ADEMP framework:

Aims: The third goal of this project is to evaluate the risk score model's performance using simulated NHANES data across three correlation scenarios: 1) observed Framingham correlation, 2) no correlation, and 3) high correlation (0.6), with the model originally developed from Framingham data.

Data-Generating Mechanisms: The data-generating process for the simulated NHANES data is informed by both the distribution shape from the Framingham dataset and the summary statistics table from the actual NHANES dataset.

Below are the distribution plots for each covariate in the Framingham dataset used for risk score prediction model development:



As we can observe, continuous variables such as TOTCHOL, SYSBP, and HDLC are normally distributed, with some skewness to the left. However, the variable AGE, which always has high density and can only be a positive number, should use a uniform distribution instead of a normal distribution for NHANES data generation. Other categorical variables like DIABETES, CURSMOKE, and BPMEDS can be generated using a binomial distribution.

The mean and standard deviation information stratified by sex for each covariate is informed by the summary statistics table from the NHANES data.

In the given framework, we consider three data-generating mechanisms for different correlation scenarios in both men and women models, setting the seed to 1 and iterating each simulation 200 times.

For men, the NHANES data are simulated for $n_{\text{target}} = 2105$, representing the male survey participants in the National Health and Nutritional Examination Study. The sex indicator is set as 1, indicating male participants. Continuous variables such as TOTCHOL, HDLC, and SYSBP for each participant are generated from a multivariate normal distribution with a mean of [183.10, 48.11, 126.44] and standard deviations of [41.65, 13.59, 16.83]. The correlation coefficients ρ, vary across three scenarios: 1) using the Framingham data correlation matrix, 2) assuming independence with a correlation matrix having zeros off-diagonal, and 3) assuming high correlation with off-diagonal values set to 0.6. The covariance matrix for the multivariate normal distribution is then calculated as follows:

$$\sigma = \begin{bmatrix} SD1^2 & \rho_{12} \times SD1 \times SD2 & \rho_{13} \times SD1 \times SD3 \\ \rho_{21} \times SD2 \times SD1 & SD2^2 & \rho_{23} \times SD2 \times SD3 \\ \rho_{31} \times SD3 \times SD1 & \rho_{32} \times SD3 \times SD2 & SD3^2 \end{bmatrix}$$

.

The AGE is derived from a uniform distribution as $Unif(18, 80)$; SYSBP, is generated as $\mathcal{N}(126.44, sd = 16.83)$, with the binary indicator BPMEDS generated from a binomial distribution as $Binom(1, 0.30)$. The variables SYSBP_UT and SYSBP_T represent the systolic blood pressure for individuals not on or on anti-hypertensive medication, respectively. For all these continuous variables, any values below 0 are adjusted to equal their current value plus the absolute minimum value of the covariate plus 1. The other two categorical variables, DIABETES and CURSMOKE, are generated as $Binom(1, 0.18)$ and $Binom(1, 0.20)$, respectively.

For women, the NHANES data are simulated on $n_{\text{target}} = 2205$, representing the female survey participants. The sex indicator is set as 2 for female participants. The continuous variable TOTCHOL, HDLC, and SYSBP for each participant is generated from multivariate normal distribution, with a mean of mean of [190.51, 58.10, 123.70]. and the standard deviation of [41.20, 15.68, 20.363]. The correlation coefficients ρ, vary across three scenarios: 1) using the Framingham data correlation matrix, 2) assuming independence with a correlation matrix having zeros off-diagonal, and 3) assuming high correlation with off-diagonal values set to 0.6. The covariance matrix for the multivariate normal distribution is defined as above. AGE, is derived from $Unif(18, 80)$; SYSBP, is derived from $\mathcal{N}(123.70, sd = 20.36)$, with BPMEDS generated as $Binom(1, 0.29)$. SYSBP_UT and SYSBP_T are defined similarly as for men. For all continuous variables, any values below 0 are adjusted to equal their current value plus the absolute minimum value of the covariate plus 1. The other categorical variables, DIABETES and CURSMOKE, are generated generated from $Binom(1, 0.12)$ and $Binom(1, 0.14)$, respectively.

Estimands: Our estimand is the Brier Score defined in the metrics sub-section that specifically designed for the evaluate the performance of the prediction developed in the framingham Study in the simulated NHANES dataset.

Methods: In this simulation, the method employed involves combining the Framingham test set with the simulated NHANES data to create a composite test dataset. We then apply a logistic regression model, developed using the Framingham training set, to predict the risk of cardiovascular disease. Each model version is stratified by sex. Additionally, logistic regression is used to estimate the inverse-odds weighting estimator by calculating $\Pr[S = 1 \mid X, D_{\text{test},i} = 1]$.

Performance Measures: We assessed the performance of each sex-stratified risk score prediction model in the target population by calculating the average of the modified Brier Score using the test set across 200 simulations.

Results

In our study, we consolidated the Brier score results from Analyses I and II into a comprehensive performance results table. The Brier score, a metric used to evaluate the accuracy of probabilistic predictions, ranges from 0 for perfect predictions to 1 for the least accurate.

Table 4: Brier Scores Stratified by Sex

	Actual NHANE	Simulated NHANES(observed corr)	Simulated NHANES(no corr)	Simulated NHANES(high corr)
Men Model	0.0759671	0.0813027	0.0841784	0.0765172
Women Model	0.0464766	0.0450132	0.0438482	0.0681365

In Analyses I and II, we used a modified version of the Brier score. This adjustment was necessary because both the actual NHANES and the simulated NHANES data are not random samples from the Framingham data, and these populations can exhibit significantly different data distributions.

Across all analyses, it was observed that the Brier scores for the women's risk score prediction model were consistently lower than those for the men's model. Notably, the Brier score for the men's model using simulated NHANES data was higher than that using actual NHANES data, implying that actual data more accurately predicts cardiovascular disease risk than simulated data based solely on summary statistics. Interestingly, the high correlation scenario for the men's model produces a Brier score similar to the actual data. In contrast, the women's model performs well with no and observed correlation in the simulated data, showing slightly lower Brier scores than the actual NHANES data. The high correlation scenario for the women's model leads to a significantly higher predictive error. These results suggest that the model's transferability to new populations varies with the underlying correlation structure in the data.

Conclusion

Overall, both analyses demonstrate that various target populations—actual NHANES, and simulated NHANES under different simulation scenarios—exhibit high accuracy and strong performance in models used for predicting cardiovascular disease risk. These findings underscore the importance of employing appropriate evaluation metrics when transporting models from a source population to assess their performance in a target population. The results also highlight the limitations inherent in using simulated data due to the variable correlation structure. Additionally, the project reveals that predictive models may perform differently across sexes, suggesting a need for sex-specific considerations in model development and evaluation.

This project has several limitations. Firstly, it relies solely on one evaluation metric, which may not fully capture the true performance of the risk score model. Researchers should consider using a modified version of the Area Under the Curve (AUC), as proposed by Li et al.[5], for more comprehensive evaluation. Secondly, the omission of missing values in the NHANES data could introduce bias or reduce the generalizability of the findings, suggesting a need for future studies to address this issue.

References

[1]D'Agostino Sr, R. B., Vasan, R. S., Pencina, M. J., Wolf, P. A., Cobain, M., Massaro, J. M., & Kannel, W. B. (2008). General cardiovascular risk profile for use in primary care: the Framingham Heart Study. Circulation, 117(6), 743-753.

[2] Randall P., NHANES: Data from the US National Health and Nutrition Examination Study. Retrieved from https://cran.r-project.org/web/packages/NHANES/index.html

[3] Steingrimsson, Jon A., et al. "Transporting a prediction model for use in a new target population." American Journal of Epidemiology 192.2 (2023): 296-304.

[4] Morris, Tim P., Ian R. White, and Michael J. Crowther. "Using simulation studies to evaluate statistical methods." Statistics in medicine 38.11 (2019): 2074-2102.

[5] Li, Bing, et al. "Estimating the area under the ROC curve when transporting a prediction model to a target population." Biometrics 79.3 (2023): 2382-2393.

Code Appendix:

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(message = F)
knitr::opts_chunk$set(warning = F)
library(riskCommunicator)
library(tidyverse)
library(tableone) # summary statistics table
library(MASS)
library(naniar) # miss_var_summary
library(kableExtra) # table styles
library(corrplot) #correlation plot
library(RColorBrewer)
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))
dim(framingham_df)

# Filter to each sex
framingham_df_men <- framingham_df %>% filter(SEX == 1)
framingham_df_women <- framingham_df %>% filter(SEX == 2)

# 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= framingham_df_men, family= "binomial")


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

```
# The NHANES data here finds the same covariates among this national survey data
library(nhanesA)

# blood pressure, demographic, bmi, smoking, and hypertension info
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")

CreateTableOne(data = df_2017, strata = c("SEX"))


# Summary table for framingham
CreateTableOne(data = framingham_df[,-c(6,11)], strata = c("SEX"))
# Correlation heat map
```

```
framingham_df_men_cor <- framingham_df_men %>%
  dplyr::select(CVD, TOTCHOL, CURSMOKE, DIABETES, HDLC, BPMEDS, SYSBP)

framingham_df_men_cor$CVD <- as.factor(framingham_df_men_cor$CVD)
framingham_df_men_cor$CURSMOKE <- as.factor(framingham_df_men_cor$CURSMOKE)
framingham_df_men_cor$DIABETES <- as.factor(framingham_df_men_cor$DIABETES)
framingham_df_men_cor$BPMEDS <- as.factor(framingham_df_men_cor$BPMEDS)

numeric_columns <- sapply(framingham_df_men_cor, is.numeric)
framingham_df_men_cnt <- framingham_df_men_cor[, numeric_columns]
cor1 <-cor(framingham_df_men_cnt, use = "complete.obs")
framingham_df_women_cor <- framingham_df_women %>%
  dplyr::select(CVD, TOTCHOL, CURSMOKE, DIABETES, HDLC, BPMEDS, SYSBP)

framingham_df_women_cor$CVD <- as.factor(framingham_df_women_cor$CVD)
framingham_df_women_cor$CURSMOKE <- as.factor(framingham_df_women_cor$CURSMOKE)
framingham_df_women_cor$DIABETES <- as.factor(framingham_df_women_cor$DIABETES)
framingham_df_women_cor$BPMEDS <- as.factor(framingham_df_women_cor$BPMEDS)

numeric_columns <- sapply(framingham_df_women_cor, is.numeric)
framingham_df_women_cnt <- framingham_df_women_cor[, numeric_columns]
cor2 <-cor(framingham_df_women_cnt, use = "complete.obs")
par(mfrow=c(1, 2))
corrplot(cor1, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
title("Correlation Plot for men")
corrplot(cor2, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
title("Correlation Plot for women")
par(mfrow=c(1, 1))

library(naniar)
# Missingness table in df_2017 Dataset
missing_names <- colnames(df_2017)[colSums(is.na(df_2017)) > 0]
df_2017_missing <- df_2017[,c(missing_names)]
df_2017_miss <- miss_var_summary(df_2017_missing)

df_2017_miss <- df_2017_miss %>% mutate(across(3, ~paste0(round(., 2), "%"))) %>% rename(Variable = var


df_2017_miss %>%
kable(booktabs = T, escape = T, caption = "Missingness in NHANES Dataset", align = "c")
# Filter to each sex
framingham_df_men <- framingham_df %>% filter(SEX == 1)
framingham_df_women <- framingham_df %>% filter(SEX == 2)
#make this example reproducible
set.seed(1)

#use 70% of dataset as training set and 30% as test set
sample_men <- sample(c(TRUE, FALSE), nrow(framingham_df_men), replace=TRUE, prob=c(0.7,0.3))
train_men  <- framingham_df_men[sample_men, ]
test_men   <- framingham_df_men[!sample_men, ]
```

```r
sample_women <- sample(c(TRUE, FALSE), nrow(framingham_df_women), replace=TRUE, prob=c(0.7,0.3))
train_women  <- framingham_df_women[sample_women, ]
test_women   <- framingham_df_women[!sample_women, ]

train_men$CURSMOKE <- as.factor(train_men$CURSMOKE)
train_men$DIABETES <- as.factor(train_men$DIABETES)

test_women$CURSMOKE <- as.factor(test_women$CURSMOKE)
test_women$DIABETES <- as.factor(test_women$DIABETES)
# train the prediction models by sex
mod_train_men <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+factor(CURSMOKE)+factor(DIABETES),
      data= train_men, family= "binomial")

mod_train_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+factor(CURSMOKE)+factor(DIABETES),
               data= train_women, family= "binomial")
predicted_men <- predict(mod_train_men, test_men, type = "response")
predicted_women <- predict(mod_train_women, test_women, type = "response")
library(DescTools)
framingham_bs_men <- BrierScore(test_men, predicted_men)
framingham_bs_women <-BrierScore(test_women, predicted_women)
# Get blood pressure based on whether or not on BPMEDS
df_2017$SYSBP_UT <- ifelse(df_2017$BPMEDS == 0, df_2017$SYSBP, 0)

df_2017$SYSBP_T <- ifelse(df_2017$BPMEDS == 1, df_2017$SYSBP, 0)
# Stratify the NHANES data by sex
train_men <- subset(train_men, select = -c(DIABP))
train_women <- subset(train_women, select = -c(DIABP))
test_men <- subset(test_men, select = -c(DIABP))
test_women <- subset(test_women, select = -c(DIABP))
train_men$S <- 1
train_women$S <- 1
test_men$S <- 1
test_women$S <- 1

df_2017_com<- subset(df_2017, select = -c(SEQN))
df_2017_com$S <- 0
df_2017_com <- na.omit(df_2017_com)
df_2017_com$CVD <- NA

df_2017_men <- df_2017_com %>% filter(SEX == 1)
df_2017_women <- df_2017_com %>% filter(SEX == 2)

# Combine the Fram test data with NHANES data by sex
test_men_comp <- rbind(test_men, df_2017_men)
test_women_comp <- rbind(test_women, df_2017_women)
test_men_comp$CURSMOKE <- as.factor(test_men_comp$CURSMOKE)
test_men_comp$DIABETES <- as.factor(test_men_comp$DIABETES)

test_women_comp$CURSMOKE <- as.factor(test_women_comp$CURSMOKE)
test_women_comp$DIABETES <- as.factor(test_women_comp$DIABETES)
# Make prediction on target population
```

```
test_men_comp$predicted_comp_men <- predict(mod_train_men, test_men_comp, type = "response")
test_women_comp$predicted_comp_women <- predict(mod_train_women, test_women_comp, type = "response")
# Calculate inverse odd weights
fit_men <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = test_men_comp, family = "binomial")
pred.prob.men <- predict(fit_men, type = "response")
test_men_comp$inv_odd_weight_men <- (1-pred.prob.men)/pred.prob.men

fit_women <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = test_women_comp, family = "binomial")
pred.prob.women <- predict(fit_women, type = "response")
test_women_comp$inv_odd_weight_women <- (1-pred.prob.women)/pred.prob.women
# Calculate brier scores
brierscore_men <- sum(test_men_comp$inv_odd_weight_men[test_men_comp$S==1] * (test_men_comp$CVD[test_men

brierscore_women <- sum(test_women_comp$inv_odd_weight_women[test_women_comp$S==1] * (test_women_comp$CV
# Create distribution plots
library(gridExtra)
a1 <- framingham_df[,-c(6,11)] %>%
ggplot(aes(x = TOTCHOL)) + geom_density(fill = "pink3") +
  labs(title = "Distribution of Total Cholesterol", x = "Total Cholesterol", y = "Density")


a2 <- framingham_df[,-c(6,11)] %>%
  ggplot(aes(x = AGE)) + geom_density(fill = "pink3") +
  labs(title = "Distribution of Age", x = "Age", y = "Density")

a3 <- framingham_df[,-c(6,11)] %>%
  ggplot(aes(x = SYSBP)) + geom_density(fill = "pink3") +
  labs(title = "Distribution of Systolic Blood Pressure", x = "Systolic Blood Pressure", y = "Count")



a4 <- framingham_df[,-c(6,11)] %>%
  ggplot(aes(x = CURSMOKE)) + geom_bar(fill = "pink3") +
  labs(title = "Distribution of Current Smoke", x = "Current Smoke", y = "Count")

a5 <- framingham_df[,-c(6,11)] %>%
  ggplot(aes(x = DIABETES)) + geom_bar(fill = "pink3")+
  labs(title = "Distribution of Diabetes", x = "Diabetes", y = "Count")


a6 <- framingham_df[,-c(6,11)]  %>%
ggplot(aes(x = HDLC)) + geom_density(fill = "pink3") +
  labs(title = "Distribution of HDL-Cholesterol", x = "Direct HDL-Cholesterol", y = "Density")

a7 <- framingham_df[,-c(6,11)]  %>%
  ggplot(aes(x = BPMEDS)) + geom_bar(fill = "pink3") +
  labs(title = "Distribution of Anti-Hypertensive Medication", x = "Anti-Hypertensive Medication", y =

grid.arrange(a1, a2, a3, a4, a5, a6, a7, ncol=2)
library(corrplot)
#Correlation heat map
```

```r
framingham_df_men_corr <- framingham_df_men %>%
  dplyr::select(CVD, TOTCHOL, AGE, CURSMOKE, DIABETES, HDLC, BPMEDS, SYSBP)
framingham_df_women_corr  <- framingham_df_women %>%
  dplyr::select(CVD, TOTCHOL, AGE, CURSMOKE, DIABETES, HDLC, BPMEDS, SYSBP)

framingham_df_men_corr$CVD <- as.factor(framingham_df_men_corr$CVD)
framingham_df_men_corr$CURSMOKE <- as.factor(framingham_df_men_corr$CURSMOKE)
framingham_df_men_corr$DIABETES <- as.factor(framingham_df_men_corr$DIABETES)
framingham_df_men_corr$BPMEDS <- as.factor(framingham_df_men_corr$BPMEDS)

framingham_df_women_corr$CVD <- as.factor(framingham_df_women_corr$CVD)
framingham_df_women_corr$CURSMOKE <- as.factor(framingham_df_women_corr$CURSMOKE)
framingham_df_women_corr$DIABETES <- as.factor(framingham_df_women_corr$DIABETES)
framingham_df_women_corr$BPMEDS <- as.factor(framingham_df_women_corr$BPMEDS)
numeric_columns <- sapply(framingham_df_men_corr, is.numeric)
framingham_df_men_cont <- framingham_df_men_corr[, numeric_columns]
M1 <-cor(framingham_df_men_cont, use = "complete.obs")
corrplot(M1, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
numeric_columns <- sapply(framingham_df_women_corr, is.numeric)
framingham_df_women_cont <- framingham_df_women_corr[, numeric_columns]
M2 <-cor(framingham_df_women_cont, use = "complete.obs")
corrplot(M2, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
tableone <- CreateTableOne(data = df_2017_com[,-12], strata = c("SEX"))
# Simulation for men model in observed Framingham correlation scenario
sim_brierscore_men_obs <- c()
set.seed(1)

for (i in 1:200){
n_men <- 2105
# SYSBP_men <- rnorm(n_men, 126.44, 16.83)
# AGE_men <- rnorm(n_men, 50.15, 18.83)
# HDLC_men <- rnorm(n_men, 48.11, 13.59)
# TOTCHOL_men <- rnorm(n_men, 183.10, 41.65)
AGE_men <- runif(n_men, 18, 80)
CURSMOKE_men <- rbinom(n_men, 1, 0.20)
BPMEDS_men <- rbinom(n_men, 1, 0.30)
DIABETES_men <- rbinom(n_men, 1, 0.18)
mean_vec_men <- c(183.10, 48.11, 126.44)
sd_vec_men <- c(41.65, 13.59,16.83)
sd_matrix_men <- diag(sd_vec_men)
sigma_men <- sd_matrix_men %*% cor1 %*% sd_matrix_men

observed_men <- mvrnorm(n = n_men, mu = mean_vec_men, Sigma = sigma_men)
TOTCHOL_men <- observed_men[,1]
HDLC_men <- observed_men[,2]
SYSBP_men <- observed_men[,3]

SEX_men <- rep(1, n_men)
SYSBP_UT_men <- ifelse(BPMEDS_men == 0, SYSBP_men, 0)
SYSBP_T_men <- ifelse(BPMEDS_men == 1, SYSBP_men, 0)
CVD_men <- NA
```

```
S_men <- 0

TOTCHOL_men <- ifelse(TOTCHOL_men <=0, TOTCHOL_men + abs(min(TOTCHOL_men)) + 1, TOTCHOL_men)
HDLC_men <- ifelse(HDLC_men <=0, HDLC_men + abs(min(HDLC_men)) + 1, HDLC_men)
SYSBP_men <- ifelse(SYSBP_men <=0, SYSBP_men + abs(min(SYSBP_men)) + 1, SYSBP_men)

sim_df_2017_men <- data.frame(CVD_men, SYSBP_men, SEX_men, AGE_men,HDLC_men, CURSMOKE_men, BPMEDS_men, S
names(sim_df_2017_men) <- c("CVD", "SYSBP", "SEX", "AGE", "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT", "SY
sim_test_comp_men <- rbind(subset(test_men, select = -c(BMI)), sim_df_2017_men)

sim_test_comp_men$CURSMOKE <- as.factor(sim_test_comp_men$CURSMOKE)
sim_test_comp_men$DIABETES <- as.factor(sim_test_comp_men$DIABETES)

sim_test_comp_men$predicted_comp_men <- predict(mod_train_men, sim_test_comp_men, type = "response")

sim_fit_men <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_men, family = "binomial")
sim.pred.prob.men <- predict(sim_fit_men, type = "response")
sim_test_comp_men$inv_odd_weight_men <- (1-sim.pred.prob.men)/sim.pred.prob.men

sim_brierscore_men_obs[i] <- sum(sim_test_comp_men$inv_odd_weight_men[sim_test_comp_men$S==1] * (sim_te
}
# Simulation for men model in no correlation scenario
sim_brierscore_men_none <- c()
set.seed(1)

for (i in 1:200){
# Number of observations
n_men <- 2105
# SYSBP_men <- rnorm(n_men, 126.44, 16.83)
# AGE_men <- rnorm(n_men, 50.15, 18.83)
# HDLC_men <- rnorm(n_men, 48.11, 13.59)
# TOTCHOL_men <- rnorm(n_men, 183.10, 41.65)
# Data generating process
AGE_men <- runif(n_men, 18, 80)
CURSMOKE_men <- rbinom(n_men, 1, 0.20)
BPMEDS_men <- rbinom(n_men, 1, 0.30)
DIABETES_men <- rbinom(n_men, 1, 0.18)

# mean and sd vector for TOTCHOL, HDLC, and SYSBP
mean_vec_men <- c(183.10, 48.11, 126.44)
sd_vec_men <- c(41.65, 13.59, 16.83)
var_matrix <- diag(sd_vec_men^2)

observed_men <- mvrnorm(n = n_men, mu = mean_vec_men, Sigma = var_matrix)
TOTCHOL_men <- observed_men[,1]
HDLC_men <- observed_men[,2]
SYSBP_men <- observed_men[,3]

SEX_men <- rep(1, n_men)
SYSBP_UT_men <- ifelse(BPMEDS_men == 0, SYSBP_men, 0)
SYSBP_T_men <- ifelse(BPMEDS_men == 1, SYSBP_men, 0)
CVD_men <- NA
```

```r
S_men <- 0

# Adjustment to negative values
TOTCHOL_men <- ifelse(TOTCHOL_men <=0, TOTCHOL_men + abs(min(TOTCHOL_men)) + 1, TOTCHOL_men)
HDLC_men <- ifelse(HDLC_men <=0, HDLC_men + abs(min(HDLC_men)) + 1, HDLC_men)
SYSBP_men <- ifelse(SYSBP_men <=0, SYSBP_men + abs(min(SYSBP_men)) + 1, SYSBP_men)

# Create composite dataset
sim_df_2017_men <- data.frame(CVD_men, SYSBP_men, SEX_men, AGE_men,HDLC_men, CURSMOKE_men, BPMEDS_men, S
names(sim_df_2017_men) <- c("CVD", "SYSBP", "SEX", "AGE", "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT", "SYS
sim_test_comp_men <- rbind(subset(test_men, select = -c(BMI)), sim_df_2017_men)

sim_test_comp_men$CURSMOKE <- as.factor(sim_test_comp_men$CURSMOKE)
sim_test_comp_men$DIABETES <- as.factor(sim_test_comp_men$DIABETES)

# Predicted values
sim_test_comp_men$predicted_comp_men <- predict(mod_train_men, sim_test_comp_men, type = "response")

# Calculate inverse odd weights
sim_fit_men <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_men, family = "binomial")
sim.pred.prob.men <- predict(sim_fit_men, type = "response")
sim_test_comp_men$inv_odd_weight_men <- (1-sim.pred.prob.men)/sim.pred.prob.men

# Calculate brier score
sim_brierscore_men_none[i] <- sum(sim_test_comp_men$inv_odd_weight_men[sim_test_comp_men$S==1] * (sim_te
}
# Simulation for men model in high correlation scenario
sim_brierscore_men_high <- c()
set.seed(1)

for (i in 1:200){
# Number of observations
n_men <- 2105
# SYSBP_men <- rnorm(n_men, 126.44, 16.83)
# AGE_men <- rnorm(n_men, 50.15, 18.83)
# HDLC_men <- rnorm(n_men, 48.11, 13.59)
# TOTCHOL_men <- rnorm(n_men, 183.10, 41.65)
# Data generating process
AGE_men <- runif(n_men, 18, 80)
CURSMOKE_men <- rbinom(n_men, 1, 0.20)
BPMEDS_men <- rbinom(n_men, 1, 0.30)
DIABETES_men <- rbinom(n_men, 1, 0.18)

# mean and sd vector for TOTCHOL, HDLC, and SYSBP
mean_vec_men <- c(183.10, 48.11, 126.44)
sd_vec_men <- c(41.65, 13.59, 16.83)
correlation_matrix <- matrix(0.6, nrow = 3, ncol = 3)
diag(correlation_matrix) <- 1  # Diagonal elements should be 1
sd_matrix <- diag(sd_vec_men)
sigma <- sd_matrix %*% correlation_matrix %*% sd_matrix

observed_men <- mvrnorm(n = n_men, mu = mean_vec_men, Sigma = sigma)
```

```r
TOTCHOL_men <- observed_men[,1]
HDLC_men <- observed_men[,2]
SYSBP_men <- observed_men[,3]

SEX_men <- rep(1, n_men)
SYSBP_UT_men <- ifelse(BPMEDS_men == 0, SYSBP_men, 0)
SYSBP_T_men <- ifelse(BPMEDS_men == 1, SYSBP_men, 0)
CVD_men <- NA
S_men <- 0

# Adjustment for negative values
TOTCHOL_men <- ifelse(TOTCHOL_men <=0, TOTCHOL_men + abs(min(TOTCHOL_men)) + 1, TOTCHOL_men)
HDLC_men <- ifelse(HDLC_men <=0, HDLC_men + abs(min(HDLC_men)) + 1, HDLC_men)
SYSBP_men <- ifelse(SYSBP_men <=0, SYSBP_men + abs(min(SYSBP_men)) + 1, SYSBP_men)

# Create composite dataset
sim_df_2017_men <- data.frame(CVD_men, SYSBP_men, SEX_men, AGE_men,HDLC_men, CURSMOKE_men, BPMEDS_men, S
names(sim_df_2017_men) <- c("CVD", "SYSBP", "SEX", "AGE", "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT", "SYS
sim_test_comp_men <- rbind(subset(test_men, select = -c(BMI)), sim_df_2017_men)

sim_test_comp_men$CURSMOKE <- as.factor(sim_test_comp_men$CURSMOKE)
sim_test_comp_men$DIABETES <- as.factor(sim_test_comp_men$DIABETES)

# Predicted values
sim_test_comp_men$predicted_comp_men <- predict(mod_train_men, sim_test_comp_men, type = "response")

# Calculate inverse odd weights
sim_fit_men <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_men, family = "binomial")
sim.pred.prob.men <- predict(sim_fit_men, type = "response")
sim_test_comp_men$inv_odd_weight_men <- (1-sim.pred.prob.men)/sim.pred.prob.men

# Calculate brier score
sim_brierscore_men_high[i] <- sum(sim_test_comp_men$inv_odd_weight_men[sim_test_comp_men$S==1] * (sim_te
}
# Simulation for women model in observed Framingham correlation scenario
sim_brierscore_women_obs <- c()
set.seed(1)

for (i in 1:200){

n_women <- 2205
#SYSBP_women <- rnorm(n_women, 123.70, 20.36)
AGE_women <- runif(n_women, 18, 80)
#AGE_women <- rnorm(n_women, 48.90, 18.57)
#HDLC_women <- rnorm(n_women, 58.10, 15.68)
#TOTCHOL_women <- rnorm(n_women, 190.51, 41.20)
CURSMOKE_women <- rbinom(n_women, 1, 0.14)
BPMEDS_women <- rbinom(n_women, 1, 0.29)
DIABETES_women <- rbinom(n_women, 1, 0.12)

mean_vec_women <- c(190.51, 58.10, 123.70)
sd_vec_women <- c(41.20, 15.68, 20.363)
```

```
sd_matrix_women <- diag(sd_vec_women)
sigma_women <- sd_matrix_women %*% cor2 %*% sd_matrix_women

observed_women <- mvrnorm(n = n_women, mu = mean_vec_women, Sigma = sigma_women)
TOTCHOL_women <- observed_women[,1]
HDLC_women <- observed_women[,2]
SYSBP_women <- observed_women[,3]

SEX_women <- rep(2, n_women)
SYSBP_UT_women <- ifelse(BPMEDS_women == 0, SYSBP_women, 0)
SYSBP_T_women <- ifelse(BPMEDS_women == 1, SYSBP_women, 0)
CVD_women <- NA
S_women <- 0

# Adjustment to negative values
TOTCHOL_women <- ifelse(TOTCHOL_women <=0, TOTCHOL_women + abs(min(TOTCHOL_women)) + 1, TOTCHOL_women)
HDLC_women <- ifelse(HDLC_women <=0, HDLC_women + abs(min(HDLC_women)) + 1, HDLC_women)
SYSBP_women <- ifelse(SYSBP_women <=0, SYSBP_women + abs(min(SYSBP_women)) + 1, SYSBP_women)

sim_df_2017_women <- data.frame(CVD_women, SYSBP_women, SEX_women, AGE_women, HDLC_women, CURSMOKE_women
names(sim_df_2017_women) <- c("CVD", "SYSBP", "SEX", "AGE",  "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT",
sim_test_comp_women <- rbind(subset(test_women, select = -c(BMI)), sim_df_2017_women)

sim_test_comp_women$CURSMOKE <- as.factor(sim_test_comp_women$CURSMOKE)
sim_test_comp_women$DIABETES <- as.factor(sim_test_comp_women$DIABETES)


sim_test_comp_women$predicted_comp_women <- predict(mod_train_women, sim_test_comp_women, type = "respon

sim_fit_women <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_women, family = "binomial")
sim.pred.prob.women <- predict(sim_fit_women, type = "response")
sim_test_comp_women$inv_odd_weight_women <- (1-sim.pred.prob.women)/sim.pred.prob.women

sim_brierscore_women_obs[i] <- sum(sim_test_comp_women$inv_odd_weight_women[sim_test_comp_women$S==1] *
}
# Simulation for women model in no correlation scenario
sim_brierscore_women_none <- c()
set.seed(1)

for (i in 1:200){

n_women <- 2205
#SYSBP_women <- rnorm(n_women, 123.70, 20.36)
AGE_women <- runif(n_women, 18, 80)
#AGE_women <- rnorm(n_women, 48.90, 18.57)
#HDLC_women <- rnorm(n_women, 58.10, 15.68)
#TOTCHOL_women <- rnorm(n_women, 190.51, 41.20)
CURSMOKE_women <- rbinom(n_women, 1, 0.14)
BPMEDS_women <- rbinom(n_women, 1, 0.29)
DIABETES_women <- rbinom(n_women, 1, 0.12)

mean_vec_women <- c(190.51, 58.10, 123.70)
```

```r
sd_vec_women <- c(41.20, 15.68, 20.363)
var_matrix <- diag(sd_vec_women^2)

observed_women <- mvrnorm(n = n_women, mu = mean_vec_women, Sigma = var_matrix )
TOTCHOL_women <- observed_women[,1]
HDLC_women <- observed_women[,2]
SYSBP_women <- observed_women[,3]

SEX_women <- rep(2, n_women)
SYSBP_UT_women <- ifelse(BPMEDS_women == 0, SYSBP_women, 0)
SYSBP_T_women <- ifelse(BPMEDS_women == 1, SYSBP_women, 0)
CVD_women <- NA
S_women <- 0

# Adjustment to negative values
TOTCHOL_women <- ifelse(TOTCHOL_women <=0, TOTCHOL_women + abs(min(TOTCHOL_women)) + 1, TOTCHOL_women)
HDLC_women <- ifelse(HDLC_women <=0, HDLC_women + abs(min(HDLC_women)) + 1, HDLC_women)
SYSBP_women <- ifelse(SYSBP_women <=0, SYSBP_women + abs(min(SYSBP_women)) + 1, SYSBP_women)

sim_df_2017_women <- data.frame(CVD_women, SYSBP_women, SEX_women, AGE_women, HDLC_women, CURSMOKE_wome
names(sim_df_2017_women) <- c("CVD", "SYSBP", "SEX", "AGE",  "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT",
sim_test_comp_women <- rbind(subset(test_women, select = -c(BMI)), sim_df_2017_women)

sim_test_comp_women$CURSMOKE <- as.factor(sim_test_comp_women$CURSMOKE)
sim_test_comp_women$DIABETES <- as.factor(sim_test_comp_women$DIABETES)


sim_test_comp_women$predicted_comp_women <- predict(mod_train_women, sim_test_comp_women, type = "respo

sim_fit_women <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_women, family = "binomial")
sim.pred.prob.women <- predict(sim_fit_women, type = "response")
sim_test_comp_women$inv_odd_weight_women <- (1-sim.pred.prob.women)/sim.pred.prob.women

sim_brierscore_women_none[i] <- sum(sim_test_comp_women$inv_odd_weight_women[sim_test_comp_women$S==1] *
}
# Simulation for women model in high correlation scenario
sim_brierscore_women_high <- c()
set.seed(1)

for (i in 1:200){

n_women <- 2205
#SYSBP_women <- rnorm(n_women, 123.70, 20.36)
AGE_women <- runif(n_women, 18, 80)
#AGE_women <- rnorm(n_women, 48.90, 18.57)
#HDLC_women <- rnorm(n_women, 58.10, 15.68)
#TOTCHOL_women <- rnorm(n_women, 190.51, 41.20)
CURSMOKE_women <- rbinom(n_women, 1, 0.14)
BPMEDS_women <- rbinom(n_women, 1, 0.29)
DIABETES_women <- rbinom(n_women, 1, 0.12)

mean_vec_women <- c(190.51, 58.10, 123.70)
```

```r
sd_vec_women <- c(41.20, 15.68, 20.363)
correlation_matrix <- matrix(0.6, nrow = 3, ncol = 3)
diag(correlation_matrix) <- 1  # Diagonal elements should be 1
sd_matrix <- diag(sd_vec_women)
sigma <- sd_matrix %*% correlation_matrix %*% sd_matrix

observed_women <- mvrnorm(n = n_women, mu = mean_vec_women, Sigma = sigma)
TOTCHOL_women <- observed_women[,1]
HDLC_women <- observed_women[,2]
SYSBP_women <- observed_women[,3]

SEX_women <- rep(2, n_women)
SYSBP_UT_women <- ifelse(BPMEDS_women == 0, SYSBP_women, 0)
SYSBP_T_women <- ifelse(BPMEDS_women == 1, SYSBP_women, 0)
CVD_women <- NA
S_women <- 0

# Adjustment to negative values
TOTCHOL_women <- ifelse(TOTCHOL_women <=0, TOTCHOL_women + abs(min(TOTCHOL_women)) + 1, TOTCHOL_women)
HDLC_women <- ifelse(HDLC_women <=0, HDLC_women + abs(min(HDLC_women)) + 1, HDLC_women)
SYSBP_women <- ifelse(SYSBP_women <=0, SYSBP_women + abs(min(SYSBP_women)) + 1, SYSBP_women)

sim_df_2017_women <- data.frame(CVD_women, SYSBP_women, SEX_women, AGE_women, HDLC_women, CURSMOKE_women
names(sim_df_2017_women) <- c("CVD", "SYSBP", "SEX", "AGE",  "HDLC", "CURSMOKE", "BPMEDS", "SYSBP_UT",
sim_test_comp_women <- rbind(subset(test_women, select = -c(BMI)), sim_df_2017_women)

sim_test_comp_women$CURSMOKE <- as.factor(sim_test_comp_women$CURSMOKE)
sim_test_comp_women$DIABETES <- as.factor(sim_test_comp_women$DIABETES)


sim_test_comp_women$predicted_comp_women <- predict(mod_train_women, sim_test_comp_women, type = "respon

sim_fit_women <- glm(S ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                     log(SYSBP_T+1)+CURSMOKE+DIABETES, data = sim_test_comp_women, family = "binomial")
sim.pred.prob.women <- predict(sim_fit_women, type = "response")
sim_test_comp_women$inv_odd_weight_women <- (1-sim.pred.prob.women)/sim.pred.prob.women

sim_brierscore_women_high[i] <- sum(sim_test_comp_women$inv_odd_weight_women[sim_test_comp_women$S==1]
}
brier_data <- data.frame(
  Iteration = 1:200,  # Replace with your actual iterations or time points
  BrierScore = sim_brierscore_men # Replace with your actual Brier Scores
)

b1 <- ggplot(brier_data, aes(x = Iteration, y = BrierScore)) +
  geom_line(color="pink3") +  # Creates a line plot
  geom_point(color="pink3") +  # Adds points to each data point
  theme_minimal() +  # A minimalistic theme
  labs(
    title = "Brier Score Over Time for Men Model",
    x = "Iteration",
    y = "Brier Score"
  )
```

```
brier_data <- data.frame(
  Iteration = 1:200,  # Replace with your actual iterations or time points
  BrierScore = sim_brierscore_women # Replace with your actual Brier Scores
)

b2 <- ggplot(brier_data, aes(x = Iteration, y = BrierScore)) +
  geom_line(color = "skyblue2") +  # Creates a line plot
  geom_point(color = "skyblue2") +  # Adds points to each data point
  theme_minimal() +  # A minimalistic theme
  labs(
    title = "Brier Score Over Time for Women Model",
    x = "Iteration",
    y = "Brier Score"
  )

grid.arrange(b1, b2, ncol=2)
library(kableExtra)
sim_brierscore_men_obs <- mean(sim_brierscore_men_obs)
sim_brierscore_men_none <- mean(sim_brierscore_men_none)
sim_brierscore_men_high <- mean(sim_brierscore_men_high)
sim_brierscore_women_obs <- mean(sim_brierscore_women_obs)
sim_brierscore_women_none <- mean(sim_brierscore_women_none)
sim_brierscore_women_high <- mean(sim_brierscore_women_high)

metrics_table <- data.frame(
  "Model" = c("Men Model", "Women Model"),
  "Actual NHANES" = c(brierscore_men, brierscore_women),
  "Simulated NHANES(observed corr)" = c(sim_brierscore_men_obs, sim_brierscore_women_obs),
  "Simulated NHANES(no corr)" = c(sim_brierscore_men_none, sim_brierscore_women_none),
  "Simulated NHANES(high corr)" = c(sim_brierscore_men_high, sim_brierscore_women_high)
)

# Create and format the table using kable
metrics_table %>%
kable(
      caption = "Brier Scores Stratified by Sex",
      booktabs = TRUE,
      col.names = c("", "Actual NHANE", "Simulated NHANES(observed corr)", "Simulated NHANES(no corr)",
  kable_styling(latex_options = c("HOLD_position", "scale_down"))
```