PHP2550_Project3Revision

Keviner

2023-12-15

Transportability of Prediction Models to a target simulated population.

Background: In recent years, considerable efforts have been made to assess the performance of prediction models in target populations and to transport the model performance metrics from a source to a target population(Steingrimsson et al. 2023). This becomes particularly relevant when the target population lacks detailed outcome information, a common scenario encountered in transportability analysis. In such cases, the source dataset offers valuable insights into the predictive relationships between covariates and outcomes, while only summary statistics are available for the target population. This research addresses this gap by proposing a methodology to extend predictive models to target populations where outcome data are unavailable, relying solely on summary statistics. The goal is to simulate the target population based on the covariate information from the summary statistics and transport measures from the source to the target, contributing to improved decision-making in scenarios where outcome information is limited or unavailable for the population of interest. The proposed methodology seeks to increase the applicability of predictive models in diverse settings, facilitating the translation of insights gained from one context to another.

Method: Leveraging data from the Framingham Heart Study, this research employed a logistic risk models to predict cardiovascular heart disease risk within the NHANES population. Utilizing a generalized logistic regression model on Framingham data, inverse estimated odds were applied as weights to obtain the brier risk. Adhering to ADEMP principles, our simulation study transparently transported prediction scores from Framingham to a simulated NHANES target population. Data generation for NHANES involved known probabilities, specified population quantities and a varying correlation structure. Despite lacking long-term outcomes, a comprehensive approach integrated common variables between NHANES and Framingham, with eligibility criteria based on age and a previous history of a stroke and heart attack applied from the Framingham study for alignment. The derived inverse odds weights were subsequently employed to obtain Brier score estimands. Bias and the standard errors, served as key metrics for model evaluation on both NHANES and simulated samples.

Results: The predictive model, was trained on 70% of the complete cases from the combined NHANES and Framingham dataset, and assessed on 30%. In the NHANES data, the model exhibited a weighted Brier score of 0.19 for cardiovascular prediction in men and 0.11 in women, on the non-simulated data and very lower brier scores on simulated set. These scores, while relatively low, indicate a moderate level of accuracy in predicting the probability of cardiovascular heart disease and demonstrate good calibration. Additionally, the model achieved an AUC of 0.7795 for women and 0.7248 for men, further supporting its discriminative ability.

Conclusion: Simulations enhances our ability to make evidence-based decisions, particularly in scenarios where real world experiments may be impractical, unavailable or ethically challenging. The weighted Brier scores of the prediction model in the target populations compared to the source population reflects the major differences in the covariate distribution between the two populations. The model had a bias of slightly higher bias on non-simulated data compared to the simuulated data suggesting that the model performed better on the simulated data indicating that transportability analysis can be achieved given the scenarios where data

is generated based on the available covariate information from the target population. The results shows that the model's performance is slightly better in NHANES compared to the simulated NHANES population. The analysis provides insights into how well logistic regression models predict cardiovascular disease (CVD) in different populations.

Commonly used: National Health and Nutrition Examination Survey (NHANES)/target population and Framingham Heart Study (Framingham)/source population.

Link to the Github page: https://github.com/kasigi234/PHP2550-Project3

Introduction

Simulation studies have become a useful tool for understanding complex systems in a controlled and virtual environment (Bienstock and Heuer 2022). They're used a lot in research to help us figure out hypothetical scenarios and come up with strategies, especially when it's tough to get real data or when we're missing outcome data. Simulations have continued to be used for risk assessment, decision-making, and sparking new ideas across diverse domains(Boulesteix et al. 2020). In the world of technology and research, simulations are getting more and more popular. They help us learn and make smart choices in different areas. In the field of biostatistics, simulation studies are integral, serving as a tool for methodological evaluation and statistical model assessment (Boulesteix, Strobl, and Augustin 2020). They're used to evaluate methods and check how well statistical models work.

Prediction models find utility when applied to target populations, such as healthcare systems predicting the likelihood of cardiovascular events in individuals under medical care to identify those at higher risk (Steingrimsson et al. 2023). However, the challenge arises when the source data, used to build the prediction model, differs from the target data, often drawn from distinct samples (Steingrimsson et al. 2023).

Therefore our objective is to tailor the Framingham model to suit the NHANES population and assess its performance in both NHANES population as well as in a simulated NHANES population a process commonly known as transporting a prediction model. The simulation plays a pivotal role in this process, serving as a crucial step to enhance the model's applicability across various scenarios. Through these efforts, our aim is to contribute to the evolving field of predictive modeling, utilizing simulation studies to improve the adaptability and effectiveness of prediction models in real-world situations.

This collaborative study involves Dr. Jon Steingrimsson from the Biostatistics Department at Brown University. The overarching goal of the study is to investigate the performance of a prediction model in a target population different from the one where the model was originally developed and evaluated. Our focus is on assessing the cardiovascular risk prediction model in the target population underlying the NHANES sample. Additionally, we conduct evaluations where the target population is simulated based on covariate information.

Data

The analysis employed data from 2539 participants in the Framingham Heart Study (1094 men and 1445 women) and 2838 individuals in the NHANES sample who met the eligibility criteria established by the Framingham dataset. Inclusion criteria encompassed individuals aged between 30 and 62 with a prior history of a stroke and a heart attack. The source data from Framingham and the NHANES were integrated, with 70% of the combined dataset allocated for training and the remaining 30% as the validation set.

Missing Data

Table 1: Missing Data in Target Population

	Variable	Missing Count	% Proportion
SYSBP	SYSBP	448	15.79
HDLC	HDLC	306	10.78
TOTCHOL	TOTCHOL	306	10.78
BPMEDS	BPMEDS	184	6.48
BMI	BMI	163	5.74
DIABETES	DIABETES	1	0.04

As illustrated in Table 1, the observed missing data within the target population (NHANES) shows a missing at random pattern, as indicated by the proportions of missing together for some of the variables. Notably, the variable `SYSBP` has the highest missing count at 15.79%, implying a potential systematic pattern in non response . Similarly, variables `HDLC`, and `TOTCHOL` also show relatively higher proportions of missing values, with `HDLC` and `TOTCHOL` missing together implying MAR in the missing pattern.

In the analysis, we considered the potential impact of MAR missingness on biasing estimates within the target population. To mitigate this, we employed the `mice` package in R for imputation. This method generated five imputed datasets, with complete estimates. The imputed datasets were then used in further analysis.

Methods

We assessed the sex specific calibration performance of a logistic regression model for prediction of cardiovascular disease (CVD) risk in Nhanes data. We used a composite data for the combined source and target populations based on common variables. We included an indicator variable `S` to distinguish between the source (Framingham data) and the NHANES datasets in the composite dataset. A logistic regression model was used to estimate the probability of membership in the source population. Using these model we derived the inverse-odds weights and applied these weights in training the predictive logistic regression model for CVD prediction using the Framingham sample. We then estimate the model performance in the NHANES target population. The model's performance was evaluated using weighted Brier scores and mean squared errors. The sex specific models showed slightly different score on predicting on Nhanes data. The model for men had a slightly higher Brier score (0.0196) than the model for women (0.0112) on Framingham data and 0.0423 and 0.0379 on NHANES data for men and women respectively. The standard errors for both men and women were relatively small (0.0047,0.0074) suggesting less variability in the estimated probabilities.

The Brier score computation was based on the equation below as in the (Steingrimsson et al. 2023) paper.

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

Table 2: Model Evaluation on Non-Simulated Data

Metric	Model for Men	Model for Women
AUC	0.7354	0.7810
Bias	0.0047	-0.0046
Brier- Framingham	0.0180	0.0107
Brier- NHANES	0.0227	0.0134
Standard Errors	0.0069	0.0042

Prediction Logistic Models.

We adapted and evaluated prediction models for the NHANES (target) population, distinct from the Framingham (source) population where the sex-specific models were initially developed. Leveraging covariate information from the NHANES population, our approach to transporting prediction models assumed conditional independence between the outcome and population, given the covariates:

$$\frac{Pr[S=0|X, D_{\text{test}}=1]}{Pr[S=0|X, D_{\text{test}}=1]}$$

Additionally, we ensured the positivity of the likelihood of being in the Framingham population for each covariate pattern in the NHANES population. To apply the prediction model and assess its performance in the target population, we employed inverse-odds weights to obtain the brier score estimands. This involved estimating the probability of belonging to the Framingham population based on covariates.

In the evaluation phase, bias and standard errors were utilized as metrics to assess the performance of the transported prediction model. The Framingham data, serving as the source population, had the outcome information, unlike the NHANES data, which lacked this outcome. Our methodology was applied to both simulated and real data, facilitating the transportation of a prediction model for Cardiovascular Heart disease from the Framingham Heart Study to the US population of NHANES eligible individuals.

Simulations Design

The generation of simulated individual-level data for men and women assumed a normal distribution and took into account the covariance structure inherent in the target population (NHANES). The covariance structure, reflecting the interdependencies among variables, was carefully considered to ensure that the simulated data closely reflected the statistical relationships observed in the NHANES population. The simulation process was based on the covariate information given in the summary statistics presented in Table 3 for each variable. To generate a pseudo-sample that mimics the NHANES population, a simulation size of 500 for each subgroup (men and women) was chosen based on a careful consideration of computational efficiency and statistical robustness of the estimates. To ensure the reproducibility of results, two distinct random seeds were employed for each simulation.

For continuous variables, such as `SYSBP`, `AGE`, `BMI`, `HDLC`, `TOTCHOL`, `SYSBP_UT`, and `SYSBP_T`, a multivariate normal distribution was utilized. This distribution allowed for the incorporation of covariance information by specifying the covariances and variances between pairs of variables if there existed any in the target (NHANES) population. The parameter values for mean and standard deviations were derived from the summary output in Table 3, ensuring that the simulated continuous variables not only captured the individual distributions accurately but also preserved the correlations among them.

In the case of binary variables such as `CURSMOKE`, `BPMEDS`, and `DIABETES`, a binomial distribution was utilized. The proportions from the summary table were used as parameters for this distribution, ensuring that the simulated binary variables reflected the observed relationships in the NHANES population.

Simulation Results

As shown in Table 4 the simulated summary statistics closely match the summary statistics of the Nhanes dataset. Although some of the means and standard deviations of the simulated data are slightly different from the non-simulated sample which could be due to the variable transformation. The little overlap implies that the simulation's captured distribution assumptions and effectively reproduced the individual characteristics from the Nhanes dataset, indicating reliability of the simulated results in representing the underlying Nhanes data.

Table 3: Model Evaluation on Simulated Data

Metric	Model for Men	Model for Women
AUC	0.7488152	0.7685116
Bias	0.0000000	0.0000005
Brier- Simulated	0.0001856	0.0001496

Table 3. Summary Statistics for Nhanes Data
Stratified by Sex

Characteristic	**1**, N = 1,326[1]	**2**, N = 1,512[1]
SYSBP	126 (16 72, 212)	122 (19 88, 224)
AGE	47 (10 30, 62)	46 (10 30, 62)
BMI	30 (7 16, 86)	31 (8 16, 75)
HDLC	48 (15 11, 166)	58 (16 18, 178)
CURSMOKE		
0	994 (75%)	1,265 (84%)
1	332 (25%)	247 (16%)
BPMEDS		
0	960 (78%)	1,119 (78%)
1	263 (22%)	312 (22%)
TOTCHOL	194 (40 84, 431)	195 (38 94, 352)
DIABETES		
0	1,177 (89%)	1,356 (90%)
1	149 (11%)	155 (10%)
MCQ160E		
2	1,326 (100%)	1,512 (100%)
MCQ160F		
2	1,326 (100%)	1,512 (100%)

[1]Mean (SD Range); n (%)

Transportability of Prediction Models.

In this study, we conducted a transportability analysis to assess the applicability of logistic regression models for predicting the likelihood of cardiovascular disease (CVD) in simulated Nhanes population. Using the Framingham Heart Study data as a source population, we simulated diverse datasets representing NHANES covariate information. Logistic regression models were adapted for each simulation incorporating inverse-odds weights derived from the Framingham data. These weights, were computed based on predicted CVD probabilities from the Framingham logistics model to adjust for the differences between the Framingham and NHANES populations. The weights were then used to obtain the brier risk scores. The models were used to predict CVD probabilities in simulated data, and predictive accuracy was assessed using the Bias, standard errors and AUC.

Transportability Results

The Brier scores presented in Table 2 indicate comparable performance between the simulated and non-simulated data. Notably, the model predicting cardiovascular risk in men exhibited a slightly higher Brier score (0.019) compared to the model for women (0.1166), suggesting a marginally lower predictive accuracy for cardiovascular risk in men as reflected by the higher Brier score. The bias was higher for the non-simulated data compared to the simulated NHANES population. This suggests that the model did not overestimate

the estimates on the simulated data. The model for men showed better bias compared to the model for women which appeared to be underestimating the estimates on simulated data.

Limitations

Our study had a number of limitations. First, the simulations were based on assumed distributions and parameters, which may have likely introduced uncertainty in the replication of real-world populations and does not consider outliers and extreme values. Secondly, the logistic regression models used in the study relied on a specific set of covariates, which could potentially be overlooking important variables not captured by the Framingham model.Additionally, the adaptation of inverse-odds weights assumed that the relationship between predictors and CVD remains consistent across populations, which may not hold in all instances.Furthermore, the application of the Framingham model to predict CVD probabilities in simulated population assumes that the underlying risk factors and relationships are comparable between genders.Despite these limitations, our analysis provides valuable insights into the transportability of the Framingham model and its performance, which indicates that the model's measures could be transported to a different measures than the one it was built on and would still perform optimally well if better calibrated.

Conclusion

Our research found that the non-simulated data had higher bias and standard errors compared to the simulated population. The Brier score estimands for the non-simulated population were also higher than the simulated population. However, the bias in the simulated NHANES population especially for men showed that the prediction model was performing optimally in estimating the probabilities of CVD in the simulated NHANES population. This results underscores the adaptability of a prediction model from a source population to a simulated population indicating that the measures of a prediction model for the likelihood of CVD could be transported to a population that is distinct from the one it was developed on and would still be extended to a simulated scenario especially in a setup where obtaining real data might be a challenge. Overall, the prediction model measures can be transported to different populations than the one it was built and evaluated in. This analysis contributes insights into the model's robustness and generalization when applied beyond its original study population.

References

Bienstock, Jared, and Albert Heuer. 2022. "A Review on the Evolution of Simulation-Based Training to Help Build a Safer Future." *Medicine* 101 (25): e29503. https://doi.org/10.1097/MD.0000000000029503.

Boulesteix, Anne-Laure, Rolf HH Groenwold, Michal Abrahamowicz, Harald Binder, Matthias Briel, Roman Hornung, Tim P Morris, Jörg Rahnenführer, and Willi Sauerbrei. 2020. "Introduction to Statistical Simulations in Health Research." *BMJ Open* 10. https://doi.org/10.1136/bmjopen-2020-039921.

Boulesteix, Anne-Laure, Carolin Strobl, and Thomas Augustin. 2020. "Introduction to Statistical Simulations in Health Research." *BMJ Open* 10 (12): e039921. https://doi.org/10.1136/bmjopen-2020-039921.

Steingrimsson, Jon A, Constantine Gatsonis, Bing Li, and Issa J Dahabreh. 2023. "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.

Code Appendix

```
knitr::opts_chunk$set(message=FALSE,
                      warning=FALSE,
                      error=FALSE,
                      echo = FALSE,
                      fig.pos = "H" ,
                      fig.align = 'center')
# load required library
library(tidyverse)
library(kableExtra)
library(knitr)
library(pROC)
library(latex2exp)
library(ggplot2)
library(tidyr)
library(dplyr)
library(lattice)
library(riskCommunicator)
library(tableone)
library(nhanesA)
library(broom)
library(MASS)
library(gtsummary)
library(corpcor)
# load source data
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)

framingham_summary_stats <- 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)
```

```r
# 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)

mcq_2017 <- nhanes("MCQ_J") %>%
  dplyr::select(SEQN, MCQ160E, MCQ160F)

# Join data from different tables
nhanes_df <- 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") %>%
  full_join(mcq_2017, by = "SEQN") # added by keviner

# Eligibility criteria based on the framingham paper
nhanes_df <- nhanes_df %>%
  filter(AGE >= 30 & AGE <= 62) %>%
  filter(MCQ160E == 2 & MCQ160F == 2)

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

#nhanes_df$SEX <- as.factor(nhanes_df$SEX)
nhanes_df$CURSMOKE <- as.factor(nhanes_df$CURSMOKE)
nhanes_df$BPMEDS <- as.factor(nhanes_df$BPMEDS)

nhanes_df$MCQ160E <- as.factor(nhanes_df$MCQ160E)
nhanes_df$MCQ160F <- as.factor(nhanes_df$MCQ160F)
nhanes_df$DIABETES <- as.factor(nhanes_df$ DIABETES)

framingham_df$CURSMOKE <- as.factor(framingham_df$CURSMOKE)
framingham_df$BPMEDS <- as.factor(framingham_df$BPMEDS)
framingham_df$DIABETES <- as.factor(framingham_df$DIABETES)
#framingham_df$CVD <- as.factor(framingham_df$CVD)
#framingham_df$CVD <- factor(framingham_df$CVD, levels = c("0", "1"))

nhanes_summary_stats <- CreateTableOne(data = nhanes_df, strata = c("SEX"))


# Distribution of Missing Data for nhanes_df
missing_df <- data.frame(
  Variable = names(nhanes_df),
  missing_count = sapply(nhanes_df, function(x) sum(is.na(x)))
)

# Calculate percent missing
```

```r
missing_df$percent_missing <- round(missing_df$missing_count / nrow(nhanes_df) * 100, 2)

# Arrange by percent missing in descending order
missing_df <- missing_df %>%
  arrange(desc(percent_missing))

# Select only those with missing records
missing_df <- missing_df %>%
  filter(missing_count > 0)
missing_df$missing_count <- round(missing_df$missing_count, 2)
missing_df$percent_missing <- round(missing_df$percent_missing, 2)

missing_df %>%
kable(caption = "Missing Data in Target Population",
      col.names = c("Variable", "Missing Count", "% Proportion"),
        digits = 3,
        booktabs = TRUE) %>%
 kable_styling(latex_options = c("HOLD_position", "striped"),
               font_size=8)

# impute using mice package for nhanes df
nhanes_df_mice_out <- mice::mice(nhanes_df, 5, pri=F)

# Store each imputed data set
nhanes_df_imp <- vector("list",5)
for (i in 1:5){
   nhanes_df_imp[[i]] <- mice::complete(nhanes_df_mice_out,i)
   nhanes_df_imp <- nhanes_df_imp[[i]]
}
#nhanes_df_imp[[1]] # Example of accessing first imputed dataset

# Add variables from Framingham to NHANES data
  nhanes_df_imp$SYSBP_UT <- ifelse(nhanes_df_imp$BPMEDS == 0, nhanes_df_imp$SYSBP, 0)
  nhanes_df_imp$SYSBP_T <- ifelse(nhanes_df_imp$BPMEDS == 1, nhanes_df_imp$SYSBP, 0)
  nhanes_df_imp$CVD <- NA
# Initialize vectors for brier scores
brier_scores_men <- numeric(length(nhanes_df_imp))
brier_scores_women <- numeric(length(nhanes_df_imp))

for (i in 1:5) {

# Create source indicator
  framingham_df$S <- 1
  nhanes_df_imp$S <- 0

  # Find common variables in fram and nhanes and combine df
  common_vars <- intersect(names(framingham_df), names(nhanes_df_imp))
dat <- rbind(
  subset(framingham_df, select = common_vars),
  subset(nhanes_df_imp, select = common_vars)
)

  # Split to train and test on combined df
```

```
  train_idx <- sample(c(TRUE, FALSE), nrow(dat), replace=TRUE, prob=c(0.7,0.3))
train_df  <- dat[train_idx,]
test_df   <- dat[!train_idx,]

  # 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 = train_df[train_df$SEX == 1,], family="binomial")


  mod_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                    log(SYSBP_T+1)+CURSMOKE+DIABETES,
                data = train_df[train_df$SEX == 2,], family="binomial")

  # Fit the logistic regression model for the probability of membership in framingham
   logit_s <- glm(S ~ log(HDLC) + log(TOTCHOL) +
                    log(AGE) + log(SYSBP_UT+1) + log(SYSBP_T+1) +
                    CURSMOKE + DIABETES,
                  data = train_df, family= "binomial")

  # Get the probabilities
  fram_prob <- predict(logit_s, newdata = test_df, type = "response")

  # Calculate the Inverse Odds Weights
  test_df$inv_weights <- 1/(fram_prob/(1-fram_prob))

  # Framingham Test Data
  dat_men_test <- test_df[test_df$S == 1 & test_df$SEX == 1,]
  dat_women_test <- test_df[test_df$S == 1 & test_df$SEX == 2,]

  # Predict the probabilities of CVD for both men and women
  men_preds <- predict(mod_men, newdata = dat_men_test, type = "response")
  women_preds <- predict(mod_women, newdata = dat_women_test, type = "response")

  # Calculate the Brier score estimator for nhanes population for both gender
  brier_scores_men[i] <- sum(dat_men_test$inv_weights*(dat_men_test$CVD - men_preds)^2)/
    nrow(test_df[test_df$S == 0 & test_df$SEX == 1,])

  brier_scores_women[i] <- sum(dat_women_test$inv_weights*(dat_women_test$CVD - women_preds)^2)/
    nrow(test_df[test_df$S == 0 & test_df$SEX == 2,])
}

# Combine results from all imputed datasets
brier_score_men <- mean(brier_scores_men)
brier_score_women <- mean(brier_scores_women)

# Initialize vectors for brier scores
brier_scores_men2 <- numeric(length(framingham_df))
brier_scores_women2 <- numeric(length(framingham_df))

for (i in 1:5) {

# Create source indicator
```

```r
  framingham_df$S <- 1
  nhanes_df_imp$S <- 0

  # Find common variables in fram and nhanes and combine df
  common_vars <- intersect(names(framingham_df), names(nhanes_df_imp))
dat <- rbind(
  subset(framingham_df, select = common_vars),
  subset(nhanes_df_imp, select = common_vars)
)

  # Split to train and test on combined df
  train_idx <- sample(c(TRUE, FALSE), nrow(dat), replace=TRUE, prob=c(0.7,0.3))
train_df  <- dat[train_idx,]
test_df   <- dat[!train_idx,]

  # 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 = train_df[train_df$SEX == 1,], family="binomial")


  mod_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                     log(SYSBP_T+1)+CURSMOKE+DIABETES,
                data = train_df[train_df$SEX == 2,], family="binomial")

  # Fit the logistic regression model for the probability of membership in framingham
   logit_s <- glm(S ~ log(HDLC) + log(TOTCHOL) +
                    log(AGE) + log(SYSBP_UT+1) + log(SYSBP_T+1) +
                    CURSMOKE + DIABETES,
                  data = train_df, family= "binomial")

  # Get the probabilities
  fram_prob <- predict(logit_s, newdata = test_df, type = "response")

  # Calculate the Inverse Odds Weights
  test_df$inv_weights <- 1/(fram_prob/(1-fram_prob))

  # Framingham Test Data
  dat_men_test <- test_df[test_df$S == 1 & test_df$SEX == 1,]
  dat_women_test <- test_df[test_df$S == 1 & test_df$SEX == 2,]

  # Predict the probabilities of CVD for both men and women
  men_preds <- predict(mod_men, newdata = dat_men_test, type = "response")
  women_preds <- predict(mod_women, newdata = dat_women_test, type = "response")

  # Calculate the Brier score estimator for nhanes population for both gender
  brier_scores_men2[i] <- sum(dat_men_test$inv_weights*(dat_men_test$CVD - men_preds)^2)/
    nrow(test_df[test_df$S == 0 & test_df$SEX == 1,])

  brier_scores_women2[i] <- sum(dat_women_test$inv_weights*(dat_women_test$CVD - women_preds)^2)/
    nrow(test_df[test_df$S == 0 & test_df$SEX == 2,])
}
```

```r
# Combine results from all imputed datasets
brier_score_men2 <- mean(brier_scores_men2)
brier_score_women2 <- mean(brier_scores_women2)

# Get the true brier estimates based on framingham as the true pop
true_brier_men <- brier_score_men2
true_brier_women <- brier_score_men2

# Calculate bias
bias_men <- brier_score_men - true_brier_men
bias_women <- brier_score_women - true_brier_women

# Calculate standard errors
se_men <- sd(brier_scores_men2) / sqrt(length(brier_scores_men2))
se_women <- sd(brier_scores_women2) / sqrt(length(brier_scores_women2))

# Create ROC curve for women
roccurvewomen <- roc(dat_women_test$CVD, women_preds)

# ROC curve for men
roccurvemen <- roc(dat_men_test$CVD, men_preds)

AUC_women <- roccurvewomen$auc
AUC_men <- roccurvemen$auc

# Combine results for men
final_results_men <- data.frame(
  Metric = c("Brier- NHANES", "Brier- Framingham", "Bias", "Standard Errors", "AUC"),
  "Men Model" = c(brier_score_men, brier_score_men2, bias_men,se_men,AUC_men)
)

# Combine results for women
final_results_women <- data.frame(
  Metric = c("Brier- NHANES", "Brier- Framingham", "Bias","Standard Errors", "AUC"),
  "Women Model" = c(brier_score_women, brier_score_women2, bias_women, se_women, AUC_women)
)

# Combine both men and women results
final_results <- merge(final_results_men, final_results_women, by = "Metric")

final_results %>%
kable(caption = "Model Evaluation on Non-Simulated Data",
      col.names = c("Metric", "Model for Men", "Model for Women"),
        digits = 4,
        booktabs = TRUE) %>%
 kable_styling(latex_options = c("HOLD_position", "striped"),
               font_size=8)
# Create ROC curves for men and women
roc_data_men <- data.frame(
  TPR = roccurvemen$sensitivities,
  FPR = 1 - roccurvemen$specificities)

# ROC curves for women
```

```r
roc_data_women <- data.frame(
  TPR = roccurvewomen$sensitivities,
  FPR = 1 - roccurvewomen$specificities)

# ROC Plots
ggplot() +
  geom_line(data = roc_data_men, aes(x = FPR, y = TPR, color = "Men"), size = 1) +
  geom_line(data = roc_data_women, aes(x = FPR, y = TPR, color = "Women"), size = 1, alpha = 0.3) +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "gray") +
    labs(title = "ROC Curves for Predictive Model on Nhanes Population",
       x = "False Positive Rate",
       y = "True Positive Rate") +
  scale_color_manual(values = c("lightblue", "red")) +
  theme_minimal() +
  theme(legend.position = "bottom")

# Generate data for women

# Number of simulations
num_sim <- 500

# set parameters
num_corr_vars = c(1,2,3,4)
corrs = c(0, 0.3, 0.7, 0.9)
n <- 1512
means <- c(
  SYSBP = 122.30,
  AGE = 46.49,
  BMI = 30.69,
  HDLC = 57.72,
  TOTCHOL = 195.22,
  SYSBP_UT =  88.46,
  SYSBP_T = 25.36
)
sds <- c(
  SYSBP = 18.71,
  AGE = 9.85,
  BMI = 8.34,
  HDLC = 16.22,
  TOTCHOL = 38.06,
  SYSBP_UT = 52.79,
  SYSBP_T = 53.16
)

props <- c(
  MCQ160E = 100.0,
  MCQ160F = 100.0,
  CURSMOKE = 16.3,
  BPMEDS = 21.8,
  DIABETES = 10.3
)

generate_continuous <- function(n, means, sds) {
```

```r
  continuous_list <- lapply(seq_along(means), function(j) {
    m <- means[j]
    sd <- sds[j]
    df <- rnorm(n, m, sd)

    return(df)

  })

  return(continuous_list)
}

# apply correlation to continuous columns
apply_correlation <- function(continuous_list, corr_any, num_corr_vars, corr) {
  if (corr_any) {
    corr_idxs <- sample(seq_along(continuous_list), num_corr_vars)

     # Generate a covariance matrix based on the correlation coefficient
    Sigma <- matrix(corr, nrow = num_corr_vars, ncol = num_corr_vars)
    Sigma[lower.tri(Sigma)] <- Sigma[upper.tri(Sigma)]
    sim_dat <- mvrnorm(n = length(continuous_list[[1]]), mu = rep(0, num_corr_vars), Sigma = Sigma, emp

    k <- 1
    for (corr_idx in corr_idxs) {
      rnk <- rank(sim_dat[, k], ties.method = "first")
      sorted_dat <- sort(continuous_list[[corr_idx]])
      continuous_list[[corr_idx]] <- sorted_dat[rnk]
      k <- k + 1
    }
    rm(sim_dat)
  }
  return(continuous_list)
}

# generate individual data for women
generate_women <- function(n, means, sds, props, corr_any, num_corr_vars, corr) {
    CURSMOKE <- rbinom(n, 1, props["CURSMOKE"] / 100)
    BPMEDS <- rbinom(n, 1, props["BPMEDS"] / 100)
    DIABETES <- rbinom(n, 1, props["DIABETES"] / 100)

  continuous_list <- generate_continuous(n, means, sds)
  continuous_list <- apply_correlation(continuous_list, corr_any, length(continuous_list), corr)

  continuous_df <- data.frame(
    TOTCHOL = continuous_list[[1]],
    SYSBP = continuous_list[[2]],
    AGE = continuous_list[[3]],
    HDLC = continuous_list[[4]],
    SYSBP_UT = continuous_list[[5]],
    SYSBP_T = continuous_list[[6]]
  )

  women_data <- data.frame(
```

```
    SEX = rep(2, n),  # Only women (SEX = 2)
    BPMEDS = BPMEDS,
    CURSMOKE = CURSMOKE,
    DIABETES = DIABETES,
    continuous_df
  )

  return(women_data)
}

# Initialize an empty list to store simulation results
simulation_W <- lapply(1:num_sim, function(i) {

  # Simulate individual pop for women with varying correlation
  generate_women(1512, means, sds, props, TRUE, 3, 0.7)
})

## Generate Men data

# Number of simulations
num_sim <- 500

# set parameters
num_corr_vars = c(1,2,3,4)
corrs = c(0, 0.3, 0.7, 0.9)
n <- 1326
means <- c(
  SYSBP = 125.62,
  AGE = 46.65,
  BMI = 30.10,
  HDLC = 47.78,
  TOTCHOL = 194.01,
  SYSBP_UT = 93.08,
  SYSBP_T =  24.55
)

sds <- c(
  SYSBP = 16.12,
  AGE = 9.90,
  BMI = 6.68,
  HDLC = 14.74,
  TOTCHOL = 40.21,
  SYSBP_UT = 54.29,
  SYSBP_T = 51.52
)

props<- c(
  MCQ160E = 100.0,
  MCQ160F = 100.0,
  CURSMOKE = 25.7,
  BPMEDS = 21.5,
  DIABETES = 11.2
)
```

```r
# generate continuous columns
generate_continuous <- function(n, means, sds) {
  continuous_list <- lapply(seq_along(means), function(j) {
    m <- means[j]
    s <- sds[j]
    df <- rnorm(n, m, s)

    return(df)

  })

  return(continuous_list)
}

# apply correlation to continuous variables
apply_correlation <- function(continuous_list,corr_any,num_corr_vars, corr) {
  if (corr_any) {
    corr_idxs <- sample(seq_along(continuous_list), num_corr_vars)

     # Generate a covariance matrix based on the correlation coefficient
    Sigma <- matrix(corr, nrow = num_corr_vars, ncol = num_corr_vars)
    Sigma[lower.tri(Sigma)] <- Sigma[upper.tri(Sigma)]
    sim_dat <- mvrnorm(n = length(continuous_list[[1]]), mu = rep(0, num_corr_vars), Sigma = Sigma, emp

    k <- 1
    for (corr_idx in corr_idxs) {
      rnk <- rank(sim_dat[, k], ties.method = "first")
      sorted_dat <- sort(continuous_list[[corr_idx]])
      continuous_list[[corr_idx]] <- sorted_dat[rnk]
      k <- k + 1
    }
    rm(sim_dat)
  }
  return(continuous_list)
}

# generate individual data for men
generate_men <- function(n, means, sds, props, corr_any, num_corr_vars, corr) {
  CURSMOKE <- rbinom(n, 1, props["CURSMOKE"] / 100)
  BPMEDS <- rbinom(n, 1, props["BPMEDS"] / 100)
  DIABETES <- rbinom(n, 1, props["DIABETES"] / 100)

  continuous_list <- generate_continuous(n, means, sds)
  continuous_list <- apply_correlation(continuous_list, corr_any, length(continuous_list), corr)

  continuous_df <- data.frame(TOTCHOL = continuous_list[[1]],
                              SYSBP = continuous_list[[2]],
                              AGE = continuous_list[[3]],
                              HDLC = continuous_list[[4]],
                              SYSBP_UT = continuous_list[[5]],
                              SYSBP_T = continuous_list[[6]]
                              )
```

```
  men_data <- data.frame(SEX = rep(1, n),
                         BPMEDS = BPMEDS,
                         CURSMOKE = CURSMOKE,
                         DIABETES = DIABETES,
                         continuous_df
                         )

  return(men_data)

}

# Initialize an empty list to store simulation results
simulation_M <- lapply(1:num_sim, function(i) {
  # Simulate individual pop for men with varying correlation
  generate_men(1326, means, sds, props, TRUE, 3, 0.7)
})

# convert simulated list to df
simM_df <- as.data.frame(simulation_M)
simW_df <- as.data.frame(simulation_W)

# combine dfs for summary stats
simM_simW_df <- bind_rows(simM_df,simW_df)
simM_simW_df$CVD <- NA

# Initialize vectors for brier scores
brier_scores_men_sim <- numeric(length(simulation_M))
brier_scores_women_sim <- numeric(length(simulation_W))

for (i in 1:num_sim) {

# Create source indicator
  framingham_df$S <- 1
  simM_simW_df$S <- 0

  # Find common variables in fram and nhanes and combine df
  common_vars <- intersect(names(framingham_df), names(simM_simW_df))
dat <- rbind(
  subset(framingham_df, select = common_vars),
  subset(simM_simW_df, select = common_vars)
)

dat <- na.omit(dat)
  # Split to train and test on combined df
  train_idx <- sample(c(TRUE, FALSE), nrow(dat), replace=TRUE, prob=c(0.7,0.3))
train_df  <- dat[train_idx,]
test_df   <- dat[!train_idx,]

  # 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 = train_df[train_df$SEX == 1,], family="binomial")
```

```r
  mod_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                     log(SYSBP_T+1)+CURSMOKE+DIABETES,
                   data = train_df[train_df$SEX == 2,], family="binomial")

  # Fit the logistic regression model for the probability of membership in framingham
   logit_s <- glm(S ~ log(HDLC) + log(TOTCHOL) +
                    log(AGE) + log(SYSBP_UT+1) + log(SYSBP_T+1) +
                    CURSMOKE + DIABETES,
                  data = train_df, family= "binomial")

  # Get the probabilities
  fram_prob <- predict(logit_s, newdata = test_df, type = "response")

  # Calculate the Inverse Odds Weights
  test_df$inv_weights <- 1/((fram_prob)/(1-fram_prob))

  # Framingham Test Data
  dat_men_test <- test_df[test_df$S == 1 & test_df$SEX == 1,]
  dat_women_test <- test_df[test_df$S == 1 & test_df$SEX == 2,]

  # Predict the probabilities of CVD for both men and women
  men_preds <- predict(mod_men, newdata = dat_men_test, type = "response")
  women_preds <- predict(mod_women, newdata = dat_women_test, type = "response")

  # Calculate the Brier score estimator for nhanes population for both gender
  brier_scores_men_sim[i] <- sum(dat_men_test$inv_weights*(dat_men_test$CVD - men_preds)^2)/(nrow(test_

  brier_scores_women_sim[i] <- sum(dat_women_test$inv_weights*(dat_women_test$CVD - women_preds)^2)/ (n
}

# Combine results from all imputed datasets
brier_score_men_sim <- mean(brier_scores_men_sim)
brier_score_women_sim <- mean(brier_scores_women_sim)

# convert simulated list to df
simM_df <- as.data.frame(simulation_M)
simW_df <- as.data.frame(simulation_W)

# combine dfs for summary stats
simM_simW_df <- bind_rows(simM_df,simW_df)
simM_simW_df$CVD <- NA

# Initialize vectors for brier scores on framing data
brier_scores_men_sim2 <- numeric(length(framingham_df))
brier_scores_women_sim2 <- numeric(length(framingham_df))

for (i in 1:num_sim) {

# Create source indicator
  framingham_df$S <- 1
  simM_simW_df$S <- 0

  # Find common variables in fram and nhanes and combine df
```

```r
  common_vars <- intersect(names(framingham_df), names(simM_simW_df))
dat <- rbind(
  subset(framingham_df, select = common_vars),
  subset(simM_simW_df, select = common_vars)
)

dat <- na.omit(dat)
  # Split to train and test on combined df
  train_idx <- sample(c(TRUE, FALSE), nrow(dat), replace=TRUE, prob=c(0.7,0.3))
train_df  <- dat[train_idx,]
test_df   <- dat[!train_idx,]

  # 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 = train_df[train_df$SEX == 1,], family="binomial")


  mod_women <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                    log(SYSBP_T+1)+CURSMOKE+DIABETES,
                data = train_df[train_df$SEX == 2,], family="binomial")

  # Fit the logistic regression model for the probability of membership in framingham
   logit_s <- glm(S ~ log(HDLC) + log(TOTCHOL) +
                    log(AGE) + log(SYSBP_UT+1) + log(SYSBP_T+1) +
                    CURSMOKE + DIABETES,
                  data = train_df, family= "binomial")

  # Get the probabilities
  fram_prob <- predict(logit_s, newdata = test_df, type = "response")

  # Calculate the Inverse Odds Weights
  test_df$inv_weights <- 1/((fram_prob)/(1-fram_prob))

  # Framingham Test Data
  dat_men_test <- test_df[test_df$S == 1 & test_df$SEX == 1,]
  dat_women_test <- test_df[test_df$S == 1 & test_df$SEX == 2,]

  # Predict the probabilities of CVD for both men and women
  men_preds <- predict(mod_men, newdata = dat_men_test, type = "response")
  women_preds <- predict(mod_women, newdata = dat_women_test, type = "response")

  # Introduce random noise to predictions
  #men_preds <- men_preds + rnorm(length(men_preds), mean = 0, sd = 0.01)
  #women_preds <- women_preds + rnorm(length(women_preds), mean = 0, sd = 0.01)

  # Calculate the Brier score estimator for nhanes population for both gender
  brier_scores_men_sim2[i] <- sum(dat_men_test$inv_weights*(dat_men_test$CVD - men_preds)^2)/(nrow(test_

  brier_scores_women_sim2[i] <- sum(dat_women_test$inv_weights*(dat_women_test$CVD - women_preds)^2)/ (n
}

# Combine results from all imputed datasets
```

```r
brier_score_men_sim2 <- mean(brier_scores_men_sim2)
brier_score_women_sim2 <- mean(brier_scores_women_sim2)

# Get the true brier estimates based on framingham as the true pop for simulated df
true_brier_men2 <- brier_score_men_sim2
true_brier_women2 <- brier_score_women_sim2

# Calculate bias
bias_men2 <- brier_score_men_sim - true_brier_men2
bias_women2 <- brier_score_women_sim - true_brier_women2

# Calculate standard errors
se_men2 <- sd(brier_score_men_sim2) / sqrt(length(brier_score_men_sim2))
se_women2 <- sd(brier_score_women_sim2) / sqrt(length(brier_score_women_sim2))

# Create ROC curve for women
roccurvewomen2 <- roc(dat_women_test$CVD, women_preds)

# ROC curve for men
roccurvemen2 <- roc(dat_men_test$CVD, men_preds)

AUC_women2 <- roccurvewomen2$auc
AUC_men2 <- roccurvemen2$auc
# Combine results for men
final_results_men2 <- data.frame(
  Metric = c("Brier- Simulated",  "Bias",  "AUC"),
  "Men Model" = c(brier_score_men_sim, bias_men2,AUC_men2)
)


# Combine results for women
final_results_women2 <- data.frame(
  Metric = c("Brier- Simulated", "Bias","AUC"),
  "Women Model" = c(brier_score_women_sim, bias_women2, AUC_women2)
)

# Combine both men and women results
final_results <- merge(final_results_men2, final_results_women2, by = "Metric")

final_results %>%
kable(caption = "Model Evaluation on Simulated Data",
      col.names = c("Metric", "Model for Men", "Model for Women"),
        #digits = 4,
        booktabs = TRUE) %>%
 kable_styling(latex_options = c("HOLD_position", "striped"),
               font_size=8)
# summary output table for nhanes data
summary_table <- nhanes_df %>%
  select(-SEQN) %>%
  tbl_summary(missing = "no", by = SEX,
              statistic = all_continuous() ~ "{mean} ({sd} {min}, {max})")%>%
  as_gt() %>%
  gt::tab_header(title = "Table 3. Summary Statistics for Nhanes Data",
```

```
                  subtitle = "Stratified by Sex")
summary_table

roc_curve_women <- roc(dat_women_test$CVD, women_preds)
roc_auc_women <- auc(roc_curve_women)

roc_curve_men <- roc(dat_men_test$CVD, men_preds)
roc_auc_men <- auc(roc_curve_men)

# Create ROC curves for men and women
roc_data_men2 <- data.frame(
  TPR = roc_curve_men$sensitivities,
  FPR = 1 - roc_curve_men$specificities)

# ROC curves for women
roc_data_women2 <- data.frame(
  TPR = roc_curve_women$sensitivities,
  FPR = 1 - roc_curve_women$specificities)

# AUC
auc_label_men <- sprintf("Men AUC = %.2f", roc_auc_men)
auc_label_women <- sprintf("Women AUC = %.2f", roc_auc_women)

# ROC Plots
ggplot() +
  geom_line(data = roc_data_men2, aes(x = FPR, y = TPR, color = "Men"), size = 1, alpha = 0.7) +
  geom_line(data = roc_data_women2, aes(x = FPR, y = TPR, color = "Women"), size = 1, alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "gray") +
  labs(title = "ROC Curves on Simulated Data",
    x = "False Positive Rate",
    y = "True Positive Rate") +
  scale_color_manual(values = c("lightblue", "red")) +
  theme_minimal() +
  theme(legend.position = "bottom") +
  annotate("text",x = 0.5,y = 0.75,label = auc_label_men,
           color = "black", hjust = 0,vjust = 1) +
  annotate("text",x = 0.5,y = 0.45,label = auc_label_women,
           color = "red",hjust = 0,vjust = 1)
```