

Modeling Cardiovascular Disease Risk: Evaluating Predictive Performance in the NHANES Target Population Using Framingham Heart Study Data

Victoria Grase

2023-12-03

Abstract

This study aimed to compare simulation and data-based analysis for transportability purposes, assessing the efficacy of both approaches in generalizing findings from target populations to larger and more expansive populations who meet certain criteria. The goal was to evaluate how insights derived from simulation and empirical data analysis contribute to the transportability of results, particularly when extrapolating conclusions to broader segments of the overall population. Professionals in the healthcare system often rely on these multivariable risk algorithms to estimate the likelihood of specific CVD events in a target population. This investigation utilized data from the Framingham Heart Study through the risk Communicator package to construct a prediction model for assessing the risk of cardiovascular disease (CVD) events. Subsequently, it evaluated the model's performance in a target population derived from the National Health and Nutrition Examination Survey (NHANES) data using the nhanesA package. The logistic regression method was employed to assess the probability of CVD event occurrence in a binary setting within the NHANES target population, comprising 2,838 study participants. The study population, with a mean age of 46.65 years and 1,512 women, consisted of individuals who underwent routine examinations between 30 and 62 years of age and were free of any prior stroke or heart occurrences. The sex-specific multivariable risk functions, often referred to as "general CVD" algorithms, incorporated variables such as age, total and high-density lipoprotein cholesterol, systolic and diastolic blood pressure, treatment for hypertension, smoking status, body mass index (BMI), blood pressure medication usage, and diabetes status. The study aimed to comprehensively assess the predictive performance of the model within the NHANES target population. The findings and results of this evaluation show the Monte Carlo simulation, with a high-density cluster around the test Brier estimate is much higher than the brier score in the target NHANES population (0.047) and lower densities for brier estimates beyond 0.20, indicates a degree of variability in the model that is unaccounted for.

Introduction

This study is the paramount objective of assessing the transportability of predictive models—a critical facet in the realm of data analysis and model development. The ability of a predictive model to maintain its effectiveness and accuracy when deployed in a population different from the one it was originally developed on stands as a pivotal concern. The overarching goal here is to delve into the nuanced intricacies of how well a predictive model, initially trained on the Framingham Heart Study dataset, generalizes its performance to an entirely distinct population represented by the NHANES dataset.

The crux of transportability lies in ensuring that the insights and predictions derived from a model are not confined to the characteristics and idiosyncrasies of a single dataset but can be extrapolated to diverse populations. It acknowledges the dynamic nature of healthcare scenarios and the inherent variability among different demographic groups. To quantitatively assess this transportability, the study employs the Brier score, a fundamental metric that quantifies the accuracy of probabilistic predictions. By focusing on the Brier score, the research aims to provide a robust evaluation of how well the cardiovascular risk prediction models maintain their predictive power when navigating the transition from the Framingham dataset to the

NHANES dataset. Essentially, this study seeks to unravel the model’s adaptability and reliability across diverse contexts, shedding light on the broader applicability of the predictive insights generated.

At the heart of this investigation lies the `nhanesA` package, a pivotal tool empowering researchers with unfettered access to the extensive National Health and Nutrition Examination Survey (NHANES) data. Through the sophisticated capabilities of the `nhanesA` package, analysts can seamlessly delve into, scrutinize, and extract profound insights from this expansive dataset. This not only aids in enhancing the depth of understanding but also significantly contributes to evidence-based decision-making within the healthcare domain. The dataset utilized in this study encompasses a rich array of demographic, questionnaire, and examination data, providing a comprehensive foundation for nuanced exploration.

Simultaneously, the Framingham Heart Study dataset stands as a cornerstone in cardiovascular research, presenting a treasure trove of information indispensable for the development of predictive models related to cardiovascular diseases. The Framingham data is instrumental in evaluating risk prediction models, allowing researchers to gauge the likelihood of cardiovascular events based on a multitude of risk factors— a facet that this study replicates. The comparative analysis of cardiovascular risk factors between the Framingham and NHANES datasets uncovers intriguing distinctions in key variables. Noteworthy differences include mean participant age, with the Framingham study averaging 60.35 years and NHANES participants averaging 46.57 years. Disparities also manifest in blood pressure measurements, where Framingham participants exhibit higher systolic and diastolic blood pressure compared to NHANES participants) systolic blood pressure (138.96 mmHg) compared to all NHANES participants (123.79 mmHg), and diastolic blood pressure is higher in Framingham (80.67 mmHg) than in NHANES (77.94)). Total cholesterol levels (236.89 mg/dL, compared to NHANES participants with a lower mean of 194.61 mg/dL) further differentiate, emphasizing the significance of accounting for dataset-specific characteristics when interpreting cardiovascular risk profiles. Importantly, the Framingham cohort introduces the outcome variable of CVD events, adding another layer of complexity to the comparative analysis. These nuanced variations underscore the necessity of tailoring preventive strategies based on dataset-specific insights, acknowledging the intricate interplay of variables across diverse populations.

Aim

The primary aim of this study is to comprehensively evaluate the performance of a cardiovascular risk prediction model in a target population underlying NHANES (National Health and Nutrition Examination Survey). Importantly, the emphasis is placed on the assessment of model performance where we will be comparing brier estimates to help determine transportability of the model.

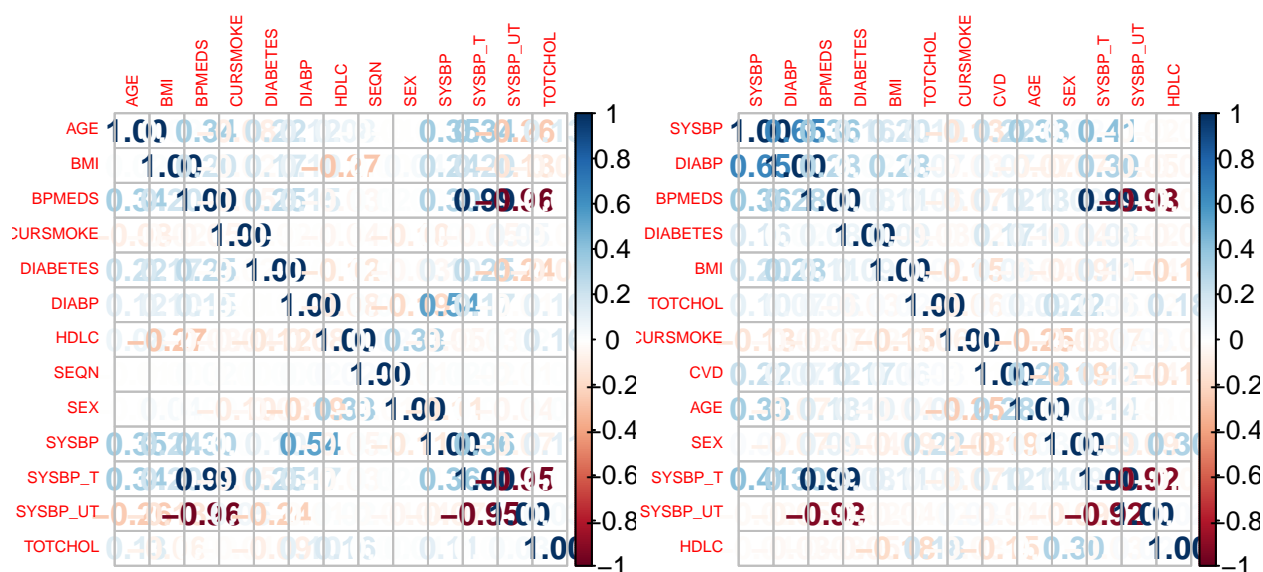
```
##                               Stratified by SEX
##                               1               2               p       test
##  n                        1110            1468
##  CVD (mean (SD))          0.32 (0.47)      0.16 (0.37)    <0.001
##  TIMECVD (mean (SD))     7226.18 (2402.62) 7952.63 (1830.88) <0.001
##  SEX (mean (SD))          1.00 (0.00)      2.00 (0.00)    <0.001
##  TOTCHOL (mean (SD))     226.34 (41.49)    246.22 (45.91)    <0.001
##  AGE (mean (SD))         60.08 (8.23)      60.62 (8.41)      0.102
##  SYSBP (mean (SD))       138.90 (21.05)    140.02 (23.74)    0.215
##  DIABP (mean (SD))       81.88 (11.41)     80.33 (11.08)     0.001
##  CURSMOKE (mean (SD))    0.39 (0.49)      0.31 (0.46)    <0.001
##  DIABETES (mean (SD))    0.09 (0.28)      0.07 (0.25)      0.049
##  BPMEDS (mean (SD))     0.11 (0.32)      0.18 (0.38)    <0.001
##  HDLC (mean (SD))        43.58 (13.36)    53.03 (15.69)    <0.001
##  BMI (mean (SD))         26.21 (3.49)      25.55 (4.25)    <0.001
```

```
## [1] 2578  14
```

```
## [1] 2539  13
```

```
##                               Stratified by SEX
##                               1               2               p       test
```

##	n	1326	1512	
##	SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001
##	TOTCHOL (mean (SD))	194.01 (40.21)	195.22 (38.06)	0.439
##	AGE (mean (SD))	46.65 (9.90)	46.49 (9.85)	0.663
##	SYSBP (mean (SD))	125.62 (16.12)	122.30 (18.71)	<0.001
##	DIABP (mean (SD))	77.80 (11.09)	73.20 (12.23)	<0.001
##	CURSMOKE (mean (SD))	0.25 (0.43)	0.16 (0.37)	<0.001
##	DIABETES (mean (SD))	0.11 (0.32)	0.10 (0.30)	0.401
##	BPMEDS (mean (SD))	0.22 (0.41)	0.22 (0.41)	0.852
##	HDLC (mean (SD))	47.78 (14.74)	57.72 (16.22)	<0.001
##	BMI (mean (SD))	30.10 (6.68)	30.69 (8.34)	0.043



In the correlation plot analysis for both the framingham and NHANES target pop, this diagonal value represents the correlation of each variable with itself, resulting in a perfect correlation. Essentially, this indicates that each variable is perfectly correlated with its own values, which is expected in a correlation matrix. Moreover, an interesting observation emerges when considering the correlation between “SYSBP” (systolic blood pressure) and “BPMEDS” (blood pressure medication usage). The correlation between these two variables is noteworthy due to the clinical context. Specifically, individuals with systolic and diastolic blood pressures exceeding 140 mmHg and 90 mmHg, respectively, are often prescribed blood pressure medications (“BPMEDS”) to manage hypertension. This relationship contributes to a positive correlation between “SYSBP” and “BPMEDS” in the correlation plot, as elevated blood pressure levels may necessitate medication. The inclusion of “SYSBP_t” (treated systolic blood pressure) and “SYSBP_ut” (untreated systolic blood pressure) further underscores the association, as they represent systolic blood pressure values under different treatment conditions.

Table 1: Brier Score Measures

Population	Brier Score
Framingham	0.16
NHANES	0.047 *estimator for Brier risk in target population

Table 2: Brier Score Measures for Men

Population	Men: Brier Score
Framingham	0.187
NHANES	0.086 *estimator for Brier risk in target population

Data Generating Mechanism

In an approach to handling missing data in the NHANES dataset, it was decided to employ multiple imputation, particularly focusing on variables DIABP (diastolic blood pressure) and SYSBP (systolic blood pressure). After implementing eligibility criteria and filtering the dataset, it was observed that DIABP and SYSBP exhibited the highest missing rates, each hovering around 15%, while the remaining variables had missingness levels below 10.7%. Given that the missing values were concentrated in specific blood pressure-related variables, and with the understanding that blood pressure measurements are critical indicators in cardiovascular health assessments, multiple imputation was deemed an appropriate strategy. By doing so, we aimed to ensure a more comprehensive and accurate representation of cardiovascular risk factors in the NHANES population, facilitating robust analyses.

The process of building a logistic regression model using the Framingham dataset for cardiovascular risk prediction and then extending the model to incorporate imputed data from the NHANES study starts when an 80/20 split is applied to the Framingham dataset for training and testing sets, respectively. Starting with the calculation of predicted probabilities for the source test set. These probabilities are computed using the logistic regression model fitted on the Framingham training data. The Brier score is then calculated for the source test set, providing a measure of the model’s accuracy by assessing the mean squared difference between predicted probabilities and actual outcomes. Subsequently, imputed data is processed by averaging across multiple datasets. Variables that are common in both the Framingham and imputed datasets are identified, and a combined dataset is formed by merging these datasets on common variables. A population indicator variable (S) is introduced to differentiate between NHANES (0) and Framingham (1) populations based on the presence or absence of participant identifiers. The final dataset is created, focusing on similar variables that facilitate meaningful comparisons between the two studies. To ensure the reproducibility of the analysis a logistic regression model is then fitted to the training set (with an 80/20 split and a set seed), incorporating key factors. Additionally, a logistic regression model specific to the Framingham population is constructed.

The logistic regression model for the combined dataset is as follows:

$$S \sim \log(HDLC + 1) + \log(TOTCHOL + 1) + \log(AGE + 1) \\ + \log(SYSBP_UT + 1) + \log(SYSBP_T + 1) + CURSMOKE + DIABETES$$

Table 3: Women Brier Score Measures

Population	Women: Brier Score
Framingham	0.142
NHANES	0.031 *estimator for Brier risk in target population

$$\hat{\psi}_{\hat{\beta}} = \frac{\sum_{i=1}^n I(S_i = 1, D_{\text{test},i} = 1) \hat{o}(X_i) (Y_i - g_{\hat{\beta}}(X_i))^2}{\sum_{i=1}^n I(S_i = 0, D_{\text{test},i} = 1)}, \quad (3)$$

where $\hat{o}(X)$ is an estimator for the inverse-odds weights in the test set, $\frac{\Pr[S = 0|X, D_{\text{test}} = 1]}{\Pr[S = 1|X, D_{\text{test}} = 1]}$.

Next we look at the computation of the Brier score, a metric that quantifies the accuracy of probability predictions made by logistic regression models. To achieve this we are looking at the function and weights above. The numerator of the Brier risk is computed by subtracting the predicted probabilities of cardiovascular disease (CVD) outcomes for the Framingham population from the observed outcomes in the combined test set. This discrepancy is squared to assess the squared differences between predicted probabilities and actual outcomes, forming the basis for the Brier score calculation. Subsequently, the weights for each observation in the combined test set are determined based on the inverse odds of being from the Framingham population ($S=1$). These weights are essential for estimating the Brier risk in the overall population, considering the NHANES population ($S=0$) as well. The weighted Brier risk is then calculated as the sum of the squared differences between observed and predicted outcomes, normalized by the total number of NHANES participants. This approach provides an estimation of the Brier risk in the broader population, acknowledging the different origins of the two study populations.

The results are summarized in Table 1,2,and 3 comparing the Brier scores for the Framingham and NHANES populations. Notably, the Framingham population exhibits a higher Brier score for overall,men,and women (0.16,0.187,0.142) suggesting a greater discrepancy between predicted probabilities and observed outcomes for the logistic regression model applied to this cohort.While, the NHANES population demonstrates a lower Brier score for overall,men, and women of (0.047,0.086,0.031), indicating a more accurate prediction based on the model. The Brier score calculated for the NHANES population serves as an estimator for the Brier risk in the broader target population, combining both Framingham and NHANES populations. The lower Brier score for NHANES suggests that the logistic regression model's predictions align more closely with observed outcomes in this population, contributing to its role as an estimator for overall Brier risk.

Estimand:

Brier Scores of a prediction model for assessing the risk of cardiovascular disease (CVD) events

Methods:

Monte Carlo Simulation:

The Monte Carlo simulation aimed to generate individual-level data for binary and continuous covariates, simulating a population that aligns with the observed NHANES data. To achieve this, we leverage two functions, each tailored to draw individual-level data under specific distributions. Firstly, for binary covariates, the normal distribution simulation function uses a Bernoulli distribution (0-1 distribution) to simulate the binary variables. The probability of success in the Bernoulli distribution is determined by the proportion of the corresponding binary covariate in the NHANES data. This approach ensures that the generated data mimics the distribution observed in NHANES. Now, for continuous covariates, two distributions are employed based on the nature of the variable. Therefore normally distributed continuous covariates, the function utilizes a normal distribution. The mean and standard deviation for each stratum are derived from the NHANES data. In addition, for positively skewed continuous covariates, the function adopts a gamma distribution. The shape and rate parameters for the gamma distribution are derived from the NHANES data. The gamma distribution is well-suited for modeling positive continuous covariates. It's to note that the decision to use these specific distributions is driven by the underlying characteristics of the covariates in the NHANES data. The Monte Carlo simulation aims to generate a synthetic population that mirrors

the observed NHANES distribution, ensuring that the simulated data is a plausible representation of the population for subsequent analyses.

Next we use the `BrierEstimate` function which requires a dataset generated by the simulation functions with parameters(`SEQN`, proportion of binary variables, mean of continuous variables, SD of the continuous variable, Sex variable for stratification, and proportion of total sample size for sex). It repeats the same steps as above for finding the brier estimate by next merging it with the original Framingham dataset, builds logistic regression models, and calculates a weighted Brier estimate for evaluating the model's performance in a combined dataset.

Next examining the distribution across the NHANES data and the simulated data from both simulation functions is important for ensuring that the generated data adequately represents the characteristics of the target population. In this context, focusing on the distribution for each gender separately is particularly important. In the context of cardiovascular disease risk prediction, where gender-specific risk factors may play a significant role, aligning the distributions becomes essential. If there are notable disparities in the distribution of key variables between the NHANES data and the simulated data, it could lead to biased models and inaccurate predictions. Therefore, a thorough examination of the distribution, particularly across genders, helps researchers identify and address any discrepancies, ensuring the robustness and applicability of their simulation results to diverse subpopulations.

Table 4: Summary of NHANES Data Proportions

SEX	CURSMOKE	DIABETES	BPMEDS
1	0.3884826	0.0877514	0.1124314
2	0.3079585	0.0657439	0.1792388

Table 5: Summary of Simulated Data Proportions-Normal Dist

SEX	CURSMOKE	DIABETES	BPMEDS
1	0.3826638	0.0916138	0.1134602
2	0.3241720	0.0634249	0.1768851

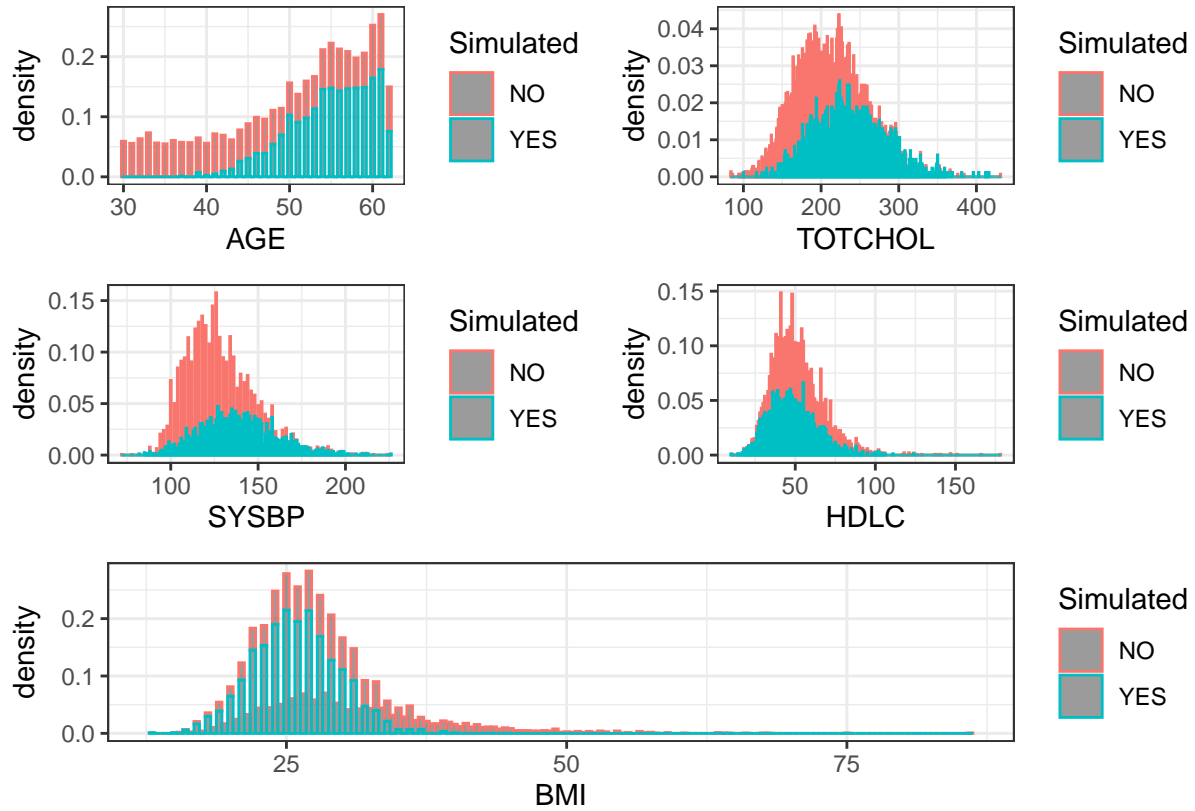


Table 6: Summary of Simulated Data Proportions-Gamma Dist

SEX	CURSMOKE	DIABETES	BPMEDS
1	0.3939394	0.0894996	0.1176885
2	0.3143058	0.0613108	0.1804087

Table 7: Summary of NHANES Data Mean

SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	60.01097	226.4442	138.9442	81.98629	43.63163	26.24624
2	60.54879	246.3183	139.9360	80.34360	53.07336	25.54848

Table 8: Summary of Simulated Data Mean-Normal Dist

SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	60.2402	228.0478	139.1479	82.35560	43.25673	26.29648
2	60.3515	244.3212	140.3824	80.27193	53.04507	25.42346

Table 9: Summary of Simulated Data Mean-Gamma Dist

SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	60.01419	226.5057	138.0643	81.67725	43.79021	26.27523
2	60.54659	247.3178	139.4392	80.21888	52.77961	25.33276

Table 10: Summary of NHANES Data SD

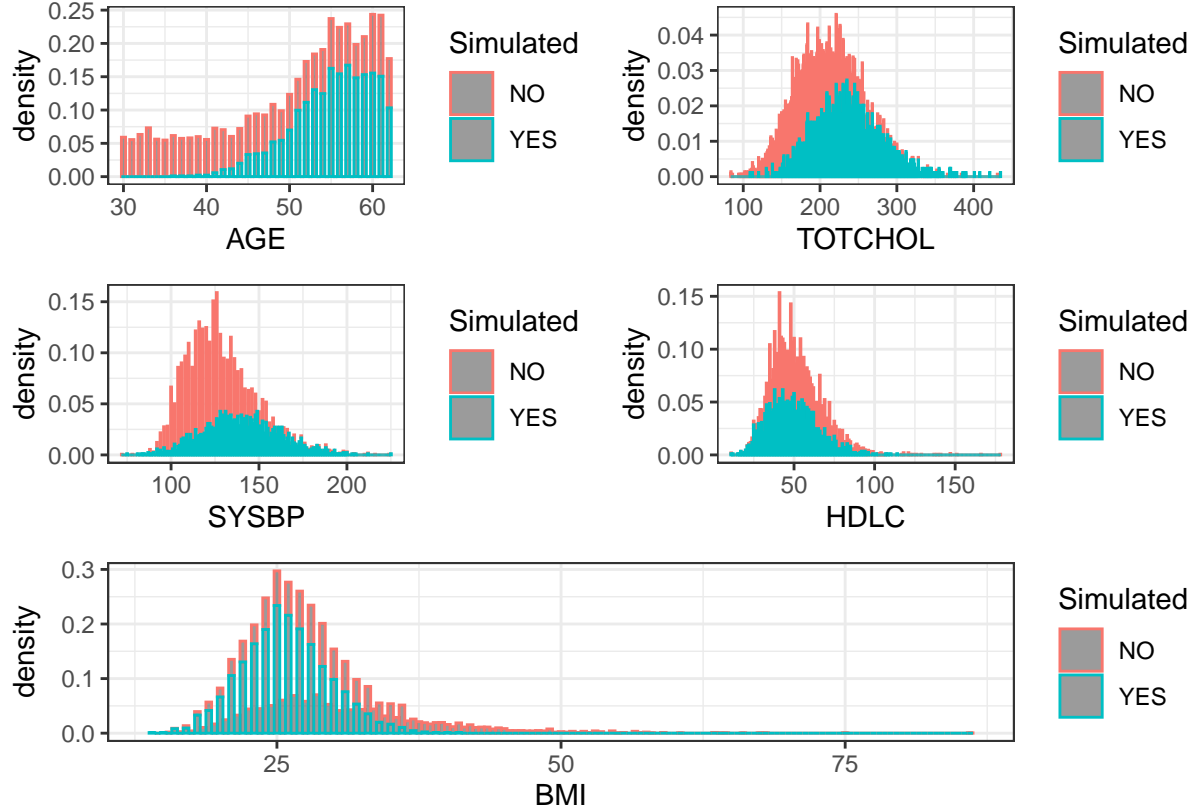
SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	8.183725	41.49070	20.88606	11.30905	13.36703	3.465203
2	8.397492	45.50815	23.70924	11.06026	15.67141	4.219409

Table 11: Summary of Simulated Data SD-Normal Dist

SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	8.018621	41.89229	21.42054	11.41931	13.06100	3.557000
2	8.322539	46.25126	22.89344	11.06292	15.17926	4.297757

Table 12: Summary of Simulated Data SD-Gamma Dist

SEX	AGE	TOTCHOL	SYSBP	DIABP	HDLC	BMI
1	8.090281	42.53585	20.86463	11.42506	13.14601	3.398382
2	8.392633	45.78922	23.86465	10.91913	15.59380	4.127709



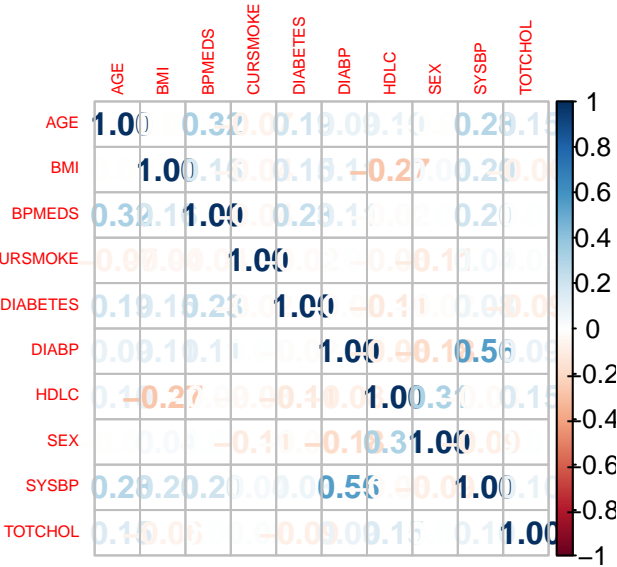
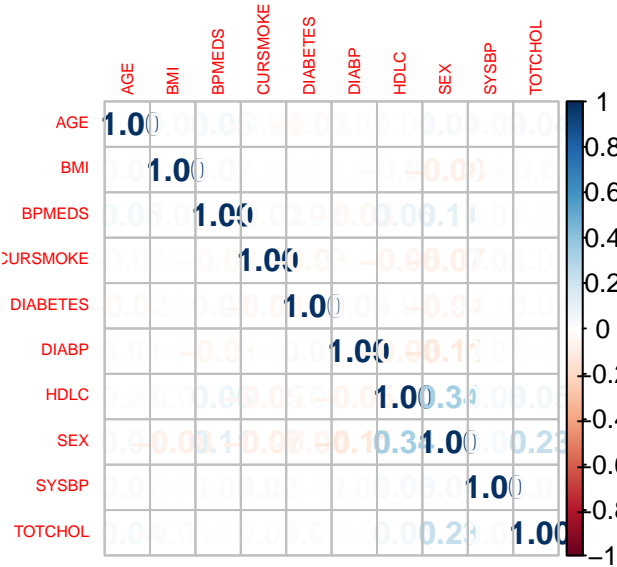
Looking at distributions we first observe the proportions (Tables:2,3,4) of smoking status, blood pressure medication, and diabetes in both of the simulated datasets closely resemble those in the NHANES data for both genders. The differences are minimal, suggesting that the simulated data adequately captures this variable's distribution.

The same thing can be said for the mean (Tables:5,6,7) and standard deviation (Tables:8,9,10) of the continuous variables: systolic blood pressure, age, HDLC, diastolic blood pressure, total cholesterol, and BMI in which the averages from the distribution closely resemble the same as the NHANES data for both genders in which you can see in the summary tables.

This is important for model transferability because when distributions are similar as they are in this case, the learned patterns, relationships, and predictive features are more likely to transfer effectively to the new population. Which is essential to check for before continuing on to see how source and target performance measures compare. Not only from the tables but you can see similar distributions in the graph. The first set representing the normal distribution and the second set of graphs modeling the Gamma distribution versus the simulated data. It is evident that the summary statistics of the simulated individual level data is similar to the summary statistics of the NHANES data.

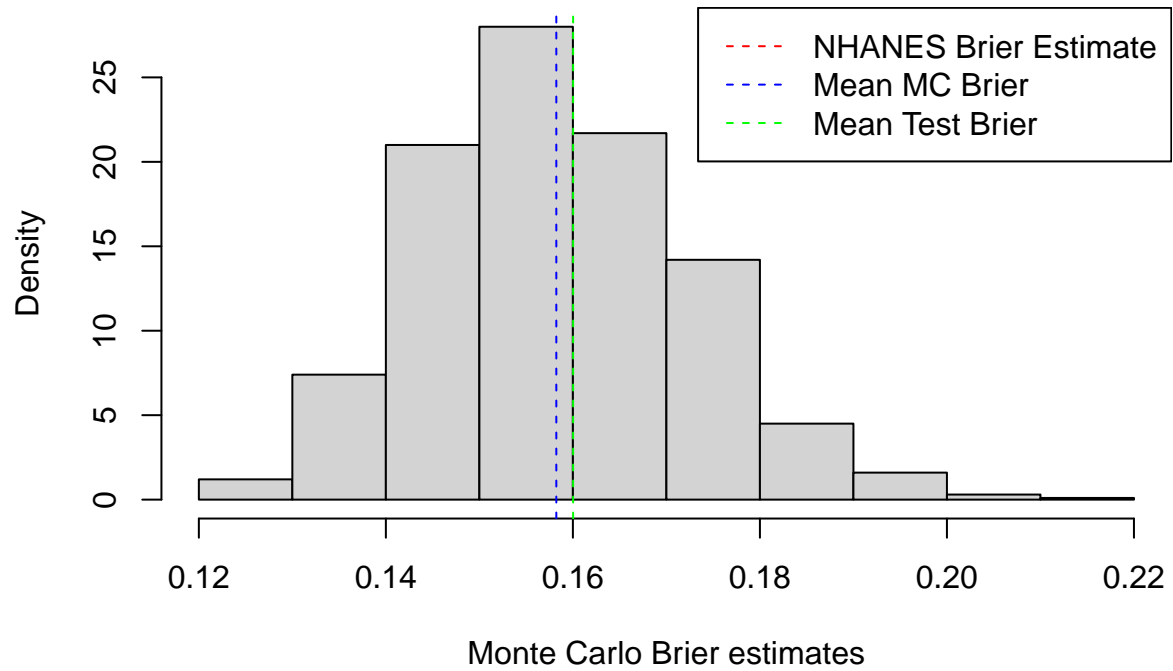
In the following, Monte Carlo Brier estimates are derived. The procedure involves two steps. (1) Simulate individual level data based on the summary statistics of the NHANES data and (2) compute the Brier estimates for each Monte Carlo draw. Before proceeding with the Monte Carlo simulations, it is essential to examine the correlation plots comparing the simulated data with the NHANES data. The diagonal values in these plots represent the correlation of each variable with itself, resulting in a perfect correlation as expected in a correlation matrix. In the NHANES imputed data, the correlation between systolic blood pressure (sysbp) and diastolic blood pressure (diabp) is observed, aligning with the physiological relationship between these two components of blood pressure. In the simulated data, a slight correlation is observed between sex and high-density lipoprotein cholesterol (HDLC). This exploration of correlations provides valuable insights into the relationships among variables and helps ensure the fidelity of the simulated data in comparison to the

NHANES dataset.

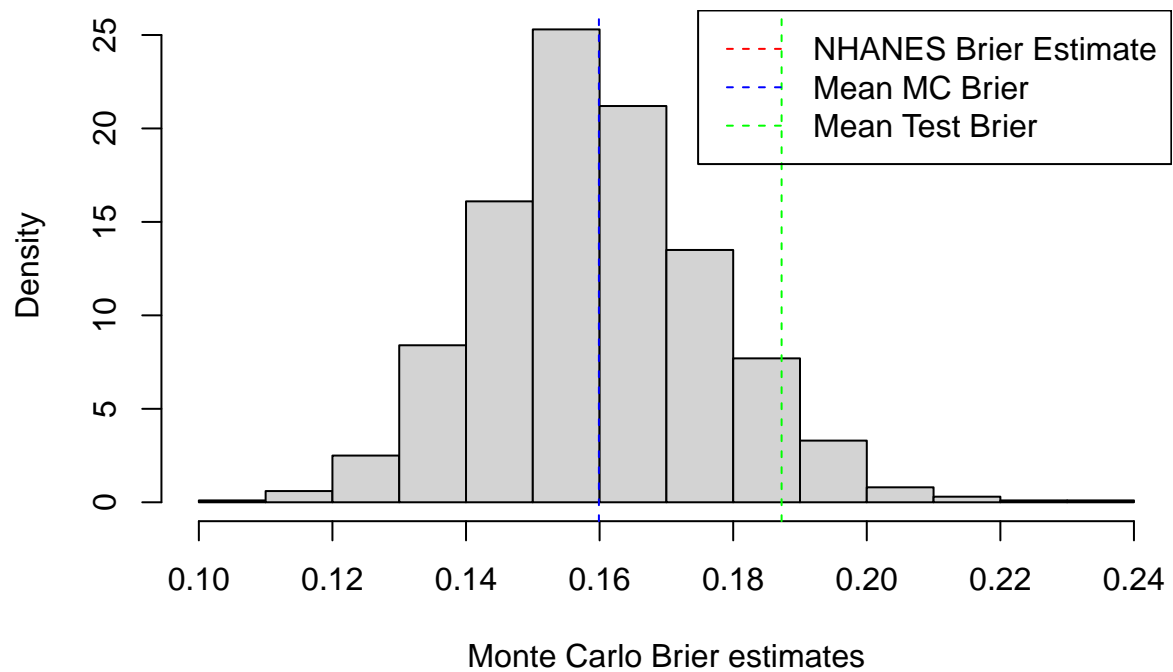


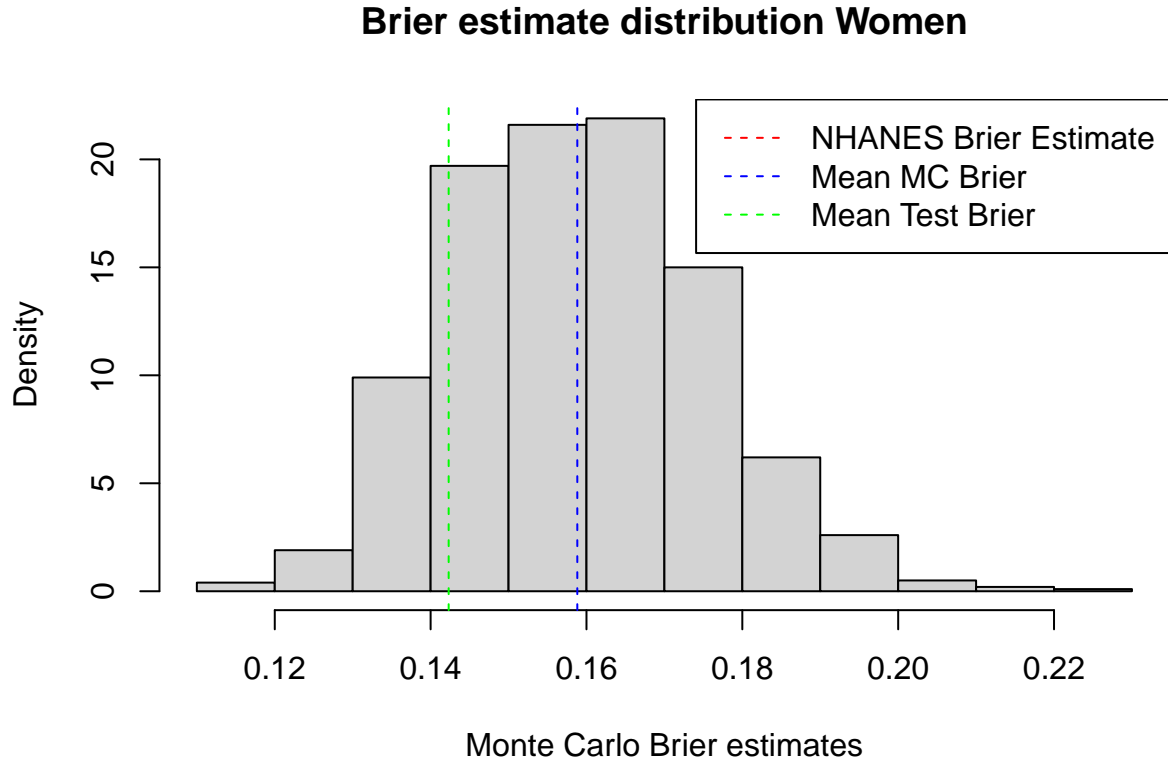
Results

Overall Brier estimate distribution



Brier estimate distribution Men





The above histogram shows the distribution of 1000 Monte Carlo Brier estimates. The vertical red line is the Brier estimate of the model on NHANES individual level data, the blue vertical line is the mean Brier estimate of the model on simulated individual level data, and the green vertical line is the brier estimate from the test set . The bar graph illustrating the Monte Carlo simulation function reveals a interesting pattern, characterized by a concentrated high density around a Brier estimate value similar to the observed estimate in the target population (0.047 in the NHANES data). This clustering implies that the simulated data faithfully mirrors the actual performance observed in the Framingham dataset but not the target population, attesting to the adjustability the simulation needs in order to capture the model's characteristics across various scenarios. The simulation's distinct density concentration around higher Brier estimates, particularly centered around 0.16. This tendency suggests a consistent inclination of the simulated data towards higher Brier scores compared to the actual estimate, indicating a lack of optimism in the model's performance. However, as Brier estimates surpass 0.20, the simulation exhibits a lower density, signifying a reduced likelihood of higher Brier scores. This nuanced observation fails to underscore the variability in simulated outcomes, not offering valuable insights into the distribution of model performance across a spectrum of Brier estimates. More information and adjustment is needed to provide a comprehensive understanding of potential scenarios, contributing to the assessment of the model's robustness in diverse settings.

Performance Measures

The utilization of the Framingham dataset with its established Brier score of (overall:0.16, men:0.187, women:0.142) as a benchmark allows for a comprehensive assessment of the model's predictive capabilities. The Monte Carlo simulation, characterized by a dense concentration around the Framingham benchmark scores while the NHANES target population Brier estimates of (overall:0.047, men:0.086, women:0.031), stands out with a significantly lower Brier score compared to the Framingham data. This notable contrast implies a potential concern in the model's predictive performance when applied to scenarios beyond the Framingham population. The simulation's consistent propensity to yield higher Brier scores, not aligning

closely with the target population, further underlines the model’s trouble targeting variability across diverse populations. This suggests that the predictive model, when subjected to the Monte Carlo simulation, does not consistently demonstrate a favorable outlook, showcasing it’s not as robust and effective in capturing the risk of cardiovascular disease events in a broader context.

Discussion

The Monte Carlo simulation, reflecting a low to none density around the Brier estimate of 0.047, 0.031, and 0.086 in the NHANES target population, signifies that this model does not generalize well across diverse populations. The simulation’s consistent alignment with the observed Framingham Brier estimate suggests that the model’s performance in the target population does not closely mirror its behavior in the Framingham dataset, providing evidence that it lacks transportability.

Furthermore, the comparison with the Framingham dataset, which serves as a benchmark with a Brier score of 0.16, 0.142, 0.187, highlights the robustness of the predictive model. The simulation consistently yields higher Brier scores, indicating a non-optimistic outlook beyond the Framingham population. This observed trend supports the notion that the model does not maintain favorable performance and predictive accuracy in scenarios beyond the original study population, substantiating its need to be revised for potential generalization.

Conclusion

The model’s lack of adaptability across different populations is a crucial aspect for healthcare professionals and policymakers, as it suggests that the risk assessment algorithm in this study is limited to specific cohorts. Instead, it needs adjustment for widespread application, catering to diverse demographics and contributing to more inclusive and accurate CVD risk assessments. Ongoing validation and scrutiny across various populations will be essential to strengthen the model’s credibility and ensure its reliable performance in real-world applications. Overall, the results underscore the potential for the predictive model to serve as a valuable tool if improved with a correct model in enhancing risk assessment strategies for cardiovascular diseases across diverse and varied populations.

Limitations

While this study employed a simulation approach using normal and gamma distributions, it’s important to acknowledge certain limitations that warrant consideration. Firstly, the choice of normal and gamma distributions for generating simulated data might not fully capture the complexity of the underlying distributions of the covariates in the real-world dataset. Different variables may exhibit diverse distributional characteristics, and utilizing a more diverse set of distributions could provide a more accurate representation of the true population.

Moreover, the logistic regression model employed in this study, while widely used for binary outcomes, is a simplification of the complex relationships that may exist within the data. More sophisticated modeling techniques, such as neural network models, could offer a more flexible representation of the underlying data structure. Neural networks are capable of capturing intricate patterns and interactions among variables, potentially enhancing the accuracy and generalizability of the predictive model.

Furthermore, the study focused on a specific set of variables for cardiovascular risk assessment, and other relevant covariates may exist that were not considered in the simulation. Exploring a broader range of predictors and their potential interactions could contribute to a more comprehensive and realistic simulation.

In conclusion, while the chosen simulation approach provides valuable insights, the study’s limitations underscore the need for ongoing refinement and exploration of alternative methodologies. Future research could delve into more diverse distributional assumptions, employ advanced modeling techniques, and expand the scope of considered covariates, ultimately enhancing the study’s applicability and robustness in capturing the complexities of cardiovascular risk assessment.

References

- 1."Transporting a prediction model for use in a new target population" (Jon A. Steingrimsson, Constantine Gatsonis, and Issa J. Dahabreh)-2021
- 2."General Cardiovascular Risk Profile for Use in Primary Care-The Framingham Heart Study" (Ralph B. D'Agostino, Sr, PhD, Ramachandran S. Vasan, MD, Michael J. Pencina, PhD, Philip A. Wolf, MD, Mark Cobain, PhD, Joseph M. Massaro, PhD, and William B. Kannel, MD)-2008
3. Li, Bing & Gatsonis, Constantine & Dahabreh, Issa & Steingrimsson, Jon. (2022). Estimating the area under the ROC curve when transporting a prediction model to a target population. *Biometrics*. 79. 10.1111/biom.13796.

Code Appendix

```
library(riskCommunicator)
library(tidyverse)
library(tableone)
library("DescTools")
library(mice)
library(knitr)
library(png)
library(VIM)
library(naniar)
library(corrplot)
library(patchwork)
library(tinytex)
library(dplyr)
library(kableExtra)
library(rmarkdown)
knitr::opts_chunk$set(echo = FALSE,
                      message = FALSE,
                      warning = FALSE)

data("framingham")

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

framingham_df <- na.omit(framingham_df)
vars <- c("CVD", "TIMECVD", "SEX", "TOTCHOL", "AGE", "SYSBP", "DIABP", "CURSMOKE", "DIABETES", "BPMEDS")
labels <- c("CVD (mean (SD))", "TIMECVD (mean (SD))", "SEX (mean (SD))",
            "Total Cholesterol (mean (SD))", "Age (mean (SD))",
            "Systolic Blood Pressure (mean (SD))", "Diastolic Blood Pressure (mean (SD))",
            "Current Smoker (mean (SD))", "Diabetes (mean (SD))",
            "Blood Pressure Medication (mean (SD))", "HDL Cholesterol (mean (SD))",
            "BMI (mean (SD))")
table_1<-CreateTableOne(data=framingham_df, strata = c("SEX"), vars = vars)
# Print the table using print instead of kable
print(table_1, label = labels, table.plain.text = FALSE)

# print(table_1, label = labels, showAllLevels = TRUE)

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)

dim(framingham_df)
framingham_df <- framingham_df %>%
  filter(!(CVD == 0 & TIMECVD <= 365*15)) %>%
  select(-c(TIMECVD))
dim(framingham_df)
```

```

framingham_df_men <- framingham_df %>% filter(SEX == 1)
framingham_df_women <- framingham_df %>% filter(SEX == 2)

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")

library(nhanesA)

bpx_2017 <- nhanes("BPX_J") %>%
  select(SEQN, BPXSY1, BPXDI1 ) %>%
  rename(SYSBP = BPXSY1, DIABP=BPXDI1)
demo_2017<- nhanes("DEMO_J") %>%
  select(SEQN, RIAGENDR, RIDAGEYR) %>%
  rename(SEX = RIAGENDR, AGE = RIDAGEYR)

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

bmx_2017 <- nhanes("BMX_J") %>%
  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)) %>%
  select(SEQN, CURSMOKE)
bpq_2017 <- nhanes("BPQ_J") %>%
  mutate(BPMEDS = case_when(
    BPQ020 == 2 ~ 0,
    BPQ040A == 2 ~ 0,
    BPQ050A == 1 ~ 1,
    TRUE ~ NA )) %>%
  select(SEQN, BPMEDS)

tchol_2017 <- nhanes("TCHOL_J") %>%
  select(SEQN, LBXTC) %>%
  rename(TOTCHOL = LBXTC)
hdl_2017 <- nhanes("HDL_J") %>%
  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)) %>%
  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(mcq_2017, by = "SEQN") %>%
  full_join(di_2017, by = "SEQN") %>%
  filter(AGE >= 30 & AGE <= 62) %>%
  filter(MCQ160F==2 & MCQ160E==2)

df_2017<-df_2017[, -c(11, 12)]

vars <- c("SEX", "TOTCHOL", "AGE", "SYSBP", "DIABP", "CURSMOKE", "DIABETES", "BPMEDS", "HDL", "BMI")
labels <- c("SEX (mean (SD))",
            "Total Cholesterol (mean (SD))", "Age (mean (SD))",
            "Systolic Blood Pressure (mean (SD))", "Diastolic Blood Pressure (mean (SD))",
            "Current Smoker (mean (SD))", "Diabetes (mean (SD))",
            "Blood Pressure Medication (mean (SD))", "HDL Cholesterol (mean (SD))",
            "BMI (mean (SD))")
table_2<-CreateTableOne(data = df_2017, strata = c("SEX"), vars=vars)

print(table_2, label = labels, table.plain.text = FALSE)

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)

par(mfrow = c(1, 2), mar = c(2, 1, 1, 1))
commonvar= interaction(colnames(framingham_df), colnames(df_2017))
df_2017 = df_2017 %>% as.data.frame() %>% select(order(commonvar))
m_NHANES= cor(na.omit(df_2017))
nhanes_corr<-corrplot(m_NHANES, method = 'number', tl.cex = 0.5)
framingham_df = framingham_df %>% as.data.frame() %>% select(order(commonvar))
m_framingham = cor(framingham_df)
fram_corr<-corrplot(m_framingham, method = 'number', tl.cex = 0.5)

nhanes_mice<- mice(df_2017, m = 5, print = FALSE, seed = 110)
nhanes_imputed<- list()
for (i in 1:5){
  nhanes_imputed[[i]]<-complete(nhanes_mice,i)
  nhanes_imputed[[i]] <- nhanes_imputed[[i]][,-1] #remove seqn
}
completed_data<-complete(nhanes_mice)

set.seed(123)
sample_fram<-sample(c(TRUE,FALSE),nrow(framingham_df), replace=TRUE, prob = c(0.8,0.2))

```

```

test_fram_17<-framingham_df[!sample_fram,]
train_fram_17<-framingham_df[sample_fram,]

model_both<- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
  log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= train_fram_17, family= "binomial")

test_prob<- predict(model_both, newdata = test_fram_17, type = "response")

test_brier<- BrierScore(as.numeric(as.character(test_fram_17$CVD)), test_prob)

imputed_data<- completed_data %>%
  group_by(SEQN) %>%
  summarise_all(mean, na.rm = TRUE)

common_vars <- intersect(names(framingham_df), names(imputed_data))

combine_df <- merge(framingham_df, imputed_data, by = common_vars, all = TRUE)

combine_df$S <- if_else(is.na(combine_df$SEQN), 1, 0)

same_vars <- c("CVD","SEX", "AGE", "TOTCHOL", "SYSBP", "CURSMOKE", "DIABETES", "BPMEDS",
  "HDLC", "BMI", "S")

combined_df <- combine_df[same_vars]

combined_df$SYSBP_UT <- ifelse(combined_df$BPMEDS == 0, combined_df$SYSBP, 0)
combined_df$SYSBP_T <- ifelse(combined_df$BPMEDS == 1, combined_df$SYSBP, 0)

set.seed(1)

sample <- sample(c(TRUE, FALSE), nrow(combined_df), replace=TRUE, prob=c(0.8,0.2))
combined_train_df <- combined_df[sample, ]
combined_test_df <- combined_df[!sample, ]

model_combined <- glm(S~log(HDLC)+log(TOTCHOL)+log(AGE)+log(BMI)+
  log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= combined_train_df, family= "binomial")

#summary(model_combined)

model_Y<- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(BMI)+
  log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= combined_train_df,
  family= "binomial")

#summary(model_Y)

combined_test_df_rm = na.omit(combined_test_df)
model_combined_predict_probs <- predict(model_combined, newdata = combined_test_df_rm, type = "response")
model_Y_predict_probs <- predict(model_Y, newdata = combined_test_df_rm, type = "response")

```

```

combined_brier<-(combined_test_df_rm$CVD-model_Y_predict_probs)^2

inv_odds_weights<-(1-model_combined_predict_probs)/(model_combined_predict_probs)

brier_estimate<-sum(combined_brier*inv_odds_weights)/sum(combined_test_df$S==0)

brierscores <- data.frame(Population = c("Framingham", "NHANES"), "Brier Score" = c(round(test_brier,3))

kable(brierscores, col.names = c("Population", "Brier Score"), caption = "Brier Score Measures", align = "center",
set.seed(123)
sample_fram_men<-sample(c(TRUE,FALSE),nrow(framingham_df_men), replace=TRUE, prob = c(0.8,0.2))
test_fram_men<-framingham_df_men[!sample_fram_men,]
train_fram_men<-framingham_df_men[sample_fram_men,]

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

test_prob_men<- predict(model_men, newdata = test_fram_men, type = "response")

test_brier_men<- BrierScore(as.numeric(as.character(test_fram_men$CVD)), test_prob_men)

imputed_men<- completed_data %>%
  group_by(SEQN) %>%
  filter(SEX == 1) %>%
  summarise_all(mean, na.rm = TRUE)

common_vars_men<- intersect(names(framingham_df_men), names(imputed_men))

combine_df_men<- merge(framingham_df_men, imputed_men, by = common_vars, all = TRUE)

combine_df_men$S <- if_else(is.na(combine_df_men$SEQN), 1, 0)

same_vars_men<- c("CVD","SEX", "AGE", "TOTCHOL", "SYSBP", "CURSMOKE", "DIABETES", "BPMEDS",
                  "HDLC", "BMI", "S")

combined_df_men<- combine_df_men[same_vars_men]

combined_df_men$SYSBP_UT <- if_else(combined_df_men$BPMEDS == 0, combined_df_men$SYSBP, 0)
combined_df_men$SYSBP_T <- if_else(combined_df_men$BPMEDS == 1, combined_df_men$SYSBP, 0)

set.seed(1)

sample2<- sample(c(TRUE, FALSE), nrow(combined_df_men), replace=TRUE, prob=c(0.8,0.2))
combined_train_df_men<- combined_df_men[sample2, ]
combined_test_df_men<- combined_df_men[!sample2, ]

model_combined_men<- glm(S~log(HDLC)+log(TOTCHOL)+log(AGE)+log(BMI)+
                        log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
                        data= combined_train_df_men, family= "binomial")

```

```

#summary(model_combined_men)

model_Y_men<- glm(CVD~log(HDLc)+log(TOTCHOL)+log(AGE)+log(BMI)+
  log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= combined_train_df_men,
  family= "binomial")

#summary(model_Y)

combined_test_df_rm2= na.omit(combined_test_df_men)
model_combined_men_predict_probs <- predict(model_combined_men, newdata = combined_test_df_rm2, type = 
model_Y_predict_probs_men<- predict(model_Y_men, newdata = combined_test_df_rm2, type = "response")

brier_men<-(combined_test_df_rm2$CVD-model_Y_predict_probs_men)^2

inv_odds_weights_men<-(1-model_combined_men_predict_probs)/(model_combined_men_predict_probs)

brier_estimate_men<-sum(brier_men*inv_odds_weights_men)/sum(combined_test_df_men$S==0)

brierscores_men<- data.frame(Population = c("Framingham", "NHANES"), "Brier Score" = c(round(test_brier,
kable(brierscores_men, col.names = c("Population", "Men: Brier Score"), caption = "Brier Score Measures
set.seed(123)
sample_fram_women<-sample(c(TRUE,FALSE),nrow(framingham_df_women), replace=TRUE, prob = c(0.8,0.2))
test_fram_women<-framingham_df_women[!sample_fram_women,]
train_fram_women<-framingham_df_women[sample_fram_women,]

model_women<- glm(CVD~log(HDLc)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
  log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data=train_fram_women, family= "binomial")

test_prob_women<- predict(model_women, newdata = test_fram_women, type = "response")

test_brier_women<- BrierScore(as.numeric(as.character(test_fram_women$CVD)), test_prob_women)

imputed_women<- completed_data %>%
  group_by(SEQN) %>%
  filter(SEX == 2) %>%
  summarise_all(mean, na.rm = TRUE)

common_vars <- intersect(names(framingham_df_women), names(imputed_women))

combine_df_women<- merge(framingham_df_women, imputed_women, by = common_vars, all = TRUE)

combine_df_women$S <- if_else(is.na(combine_df_women$SEQN), 1, 0)

same_vars_women<- c("CVD", "SEX", "AGE", "TOTCHOL", "SYSBP", "CURSMOKE", "DIABETES", "BPMEDS",
  "HDLc", "BMI", "S")

```

```

combined_df_women<- combine_df_women[same_vars_women]

combined_df_women$SYSBP_UT <- ifelse(combined_df_women$BPMEDS == 0, combined_df_women$SYSBP, 0)
combined_df_women$SYSBP_T <- ifelse(combined_df_women$BPMEDS == 1, combined_df_women$SYSBP, 0)

set.seed(1)

sample1<- sample(c(TRUE, FALSE), nrow(combined_df_women), replace=TRUE, prob=c(0.8,0.2))
combined_train_df_women<- combined_df_women[sample1, ]
combined_test_df_women<- combined_df_women[!sample1, ]

model_combined_women<- glm(S~log(HDLc)+log(TOTCHOL)+log(AGE)+log(BMI)+
                           log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
                           data= combined_train_df_women, family= "binomial")

#summary(model_combined_women)

model_Y_women<- glm(CVD~log(HDLc)+log(TOTCHOL)+log(AGE)+log(BMI)+
                    log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
                    data= combined_train_df_women,
                    family= "binomial")

#summary(model_Y)

combined_test_df_rm1= na.omit(combined_test_df_women)
model_combined_women_predict_probs <- predict(model_combined_women, newdata = combined_test_df_rm1, type="response")
model_Y_predict_probs_women<- predict(model_Y_women, newdata = combined_test_df_rm1, type = "response")

brier_women<-(combined_test_df_rm1$CVD-model_Y_predict_probs_women)^2

inv_odds_weights_women<-(1-model_combined_women_predict_probs)/(model_combined_women_predict_probs)

brier_estimate_women<-sum(brier_women*inv_odds_weights_women)/sum(combined_test_df_women$S==0)

brierscores_women<- data.frame(Population = c("Framingham", "NHANES"), "Brier Score" = c(round(test_brier, 2), round(test_brier, 2)))
kable(brierscores_women, col.names = c("Population", "Women: Brier Score"), caption = "Women Brier Score")

library(MASS)

SimulateData_NormalDist <- function(SEQN,
                                    proportion_summ,
                                    mean_summ,
                                    sd_summ,
                                    cov_matrix, # Add covariance matrix parameter
                                    strata_var = "SEX",
                                    strata_prop = 0.5) {

```

```

n <- length(SEQN)

Bin <- data.frame(SEQN = SEQN)
aux_binary <- proportion_summ %>%
  dplyr::select(-all_of(strata_var)) %>%
  colnames()
rex_sex1_prop <- proportion_summ %>%
  filter(SEX == 1) %>%
  dplyr::select(all_of(aux_binary)) %>%
  as.data.frame()
rex_sex2_prop <- proportion_summ %>%
  dplyr::filter(SEX == 2) %>%
  dplyr::select(all_of(aux_binary)) %>%
  as.data.frame()

for (k in seq_len(length(aux_binary))) {
  Bin[, aux_binary[k]] <- c(rbinom(round(n * strata_prop), size = 1, prob = rex_sex1_prop[, aux_binary[k]]),
    rbinom(round(n - n * strata_prop), size = 1, prob = rex_sex2_prop[, aux_binary[k]]))
}
Bin[, strata_var] <- c(rep(1, round(n * strata_prop)),
  rep(2, round(n * strata_prop)))

C <- data.frame(SEQN = SEQN)
aux_con <- mean_summ %>%
  dplyr::select(-all_of(strata_var)) %>%
  colnames()

rex_sex1_mean <- mean_summ %>%
  filter(SEX == 1) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()
rex_sex1_sd <- sd_summ %>%
  filter(SEX == 1) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()

rex_sex2_mean <- mean_summ %>%
  filter(SEX == 2) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()
rex_sex2_sd <- sd_summ %>%
  filter(SEX == 2) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()

for (k in seq_len(length(aux_con))) {
  C[, aux_con[k]] = c(rnorm(round(n * strata_prop), mean = rex_sex1_mean[, aux_con[k]], sd = rex_sex1_sd[, aux_con[k]]),
    rnorm(round(n - n * strata_prop), mean = rex_sex2_mean[, aux_con[k]], sd = rex_sex2_sd[, aux_con[k]]))
}

# Add correlated variables using the covariance matrix

```

```

correlated_vars <- mvrnorm(n, mu = rep(0, ncol(cov_matrix)), Sigma = cov_matrix)
colnames(correlated_vars) <- paste0("Cor_Var_", seq_len(ncol(correlated_vars)))

# Merge correlated variables into the data frame
C <- cbind(C, correlated_vars)

SimulatedData = Bin %>% left_join(C, by = "SEQN")

# Add constraint to prevent nan due to log()
SimulatedData[SimulatedData < 0] = 0.001

return(SimulatedData)
}
SimulatedData_GammaDist <- function(SEQN,
                                     proportion_summ,
                                     mean_summ,
                                     sd_summ,
                                     strata_var = "SEX",
                                     strata_prop = 0.5){

n= length(SEQN)

Bin = data.frame(SEQN = SEQN)
aux_binary <- proportion_summ %>%
  dplyr::select(-all_of(strata_var)) %>%
  colnames()
rex_sex1_prop <- proportion_summ %>%
  filter(SEX==1) %>%
  dplyr::select(all_of(aux_binary)) %>%
  as.data.frame()
rex_sex2_prop <- proportion_summ %>%
  filter(SEX==2) %>%
  dplyr::select(all_of(aux_binary)) %>%
  as.data.frame()

for (k in seq_len(length(aux_binary))) {
  Bin[,aux_binary[k]] <- c(rbinom(round(n*strata_prop),size=1, prob = rex_sex1_prop[,aux_binary[k]]
                                rbinom(round(n*strata_prop),size=1, prob = rex_sex2_prop[,aux_binary[k]]
  )
}
Bin[,strata_var] <- c(rep(1,round(n*strata_prop)),
                     rep(2,round(n*strata_prop)))

C <- data.frame(SEQN = SEQN)
aux_con<- mean_summ %>%
  dplyr::select(-all_of(strata_var)) %>%
  colnames()

rex_sex1_mean <-mean_summ %>%
  filter(SEX==1) %>%
  dplyr::select(all_of(aux_con)) %>%

```

```

    as.data.frame()
rex_sex1_sd <-sd_summ %>%
  filter(SEX==1) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()

rex_sex2_mean <-mean_summ %>%
  filter(SEX==2) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()
rex_sex2_sd<-sd_summ %>%
  filter(SEX==2) %>%
  dplyr::select(all_of(aux_con)) %>%
  as.data.frame()

transform_shape_sex1 = rex_sex1_mean^2/rex_sex1_sd^2
transform_rate_sex1 = rex_sex1_mean/rex_sex1_sd^2
# Sex==2
transform_shape_sex2 = rex_sex2_mean^2/rex_sex2_sd^2
transform_rate_sex2 = rex_sex2_mean/rex_sex2_sd^2

for (k in seq_len(length(aux_con))) {
  C[,aux_con[k]] = c(rgamma(round(n*strata_prop),shape = transform_shape_sex1[,aux_con[k]], rate =
    rgamma(round(n-n*strata_prop),shape = transform_shape_sex2[,aux_con[k]], rate =

}
SimulatedData = Bin %>% left_join(C,by="SEQN")

return(SimulatedData)
}

BrierEstimate <- function(DATA){

  common_vars <- intersect(names(framingham_df), names(DATA))

  combine_df<- merge(framingham_df, DATA, by = common_vars, all = TRUE)

  combine_df$S <- if_else(is.na(combine_df$SEQN), 1, 0)

  same_vars <- c("CVD", "SEX", "AGE", "TOTCHOL", "SYSBP", "CURSMOKE", "DIABETES", "BPMEDS",
    "HDL", "BMI", "S")

  combined_df <- combine_df[same_vars]

  combined_df$SYSBP_UT <- ifelse(combined_df$BPMEDS == 0, combined_df$SYSBP, 0)
  combined_df$SYSBP_T <- ifelse(combined_df$BPMEDS == 1, combined_df$SYSBP, 0)

```



```

sample <- sample(c(TRUE, FALSE), nrow(combined_df), replace=TRUE, prob=c(0.8,0.2))
combined_train_df <- combined_df[sample, ]
combined_test_df <- combined_df[!sample, ]

model_combined <- glm(S~log(HDLC)+log(TOTCHOL)+log(AGE)+log(BMI)+
  log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= combined_train_df, family= "binomial")

summary(model_combined)

model_Y<- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(BMI)+
  log(SYSBP_UT+1)+log(SYSBP_T+1)+CURSMOKE+DIABETES,
  data= combined_train_df,
  family= "binomial")

summary(model_Y)

combined_test_df_rm = na.omit(combined_test_df)

model_combined_predict_probs <- predict(model_combined, newdata = combined_test_df_rm, type = "response")
model_Y_predict_probs <- predict(model_Y, newdata = combined_test_df_rm, type = "response")

combined_brier <- (combined_test_df_rm$CVD-model_Y_predict_probs)^2

inv_odds_weights <- (1-model_combined_predict_probs)/(model_combined_predict_probs)

brier_estimate <- sum(combined_brier*inv_odds_weights)/sum(combined_test_df_rm$S==0)

return(brier_estimate)
}

same_vars<- c("SEX", "AGE", "TOTCHOL", "SYSBP","DIABP", "CURSMOKE", "DIABETES", "BPMEDS",
  "HDLC", "BMI")

same_vars.ContinouseDistribution=c("AGE", "TOTCHOL", "SYSBP","DIABP",
  "HDLC", "BMI")
same_vars.DiscreteDistribution=c("CURSMOKE", "DIABETES", "BPMEDS")
library(dplyr)
nhanes_summary_prop<-framingham_df[,same_vars] %>%
  group_by(SEX) %>%
  dplyr::select(all_of(same_vars.DiscreteDistribution)) %>%
  summarise_all(mean,na.rm = T)

nhanes_summary_mean <-framingham_df[,same_vars] %>%
  group_by(SEX) %>%
  dplyr::select(all_of(same_vars.ContinouseDistribution)) %>%
  summarise_all(mean,na.rm = T)

```

```

nhanes_summary_sd <- framingham_df[, same_vars] %>%
  group_by(SEX) %>%
  dplyr::select(all_of(same_vars.ContinouseDistribution)) %>%
  summarise_all(sd, na.rm = T)

# Example covariance matrix
cov_matrix <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)

DATA = SimulateData_NormalDist(SEQN = imputed_data$SEQN,
                                proportion_summ = nhanes_summary_prop,
                                mean_summ = nhanes_summary_mean,
                                sd_summ = nhanes_summary_sd,
                                cov_matrix, # Add covariance matrix parameter
                                strata_var = "SEX",
                                strata_prop = 0.5)

same_vars <- c("SEX", "AGE", "TOTCHOL", "SYSBP", "DIABP", "CURSMOKE", "DIABETES", "BPMEDS", "HDL", "BMI")
DATA_summary_prop <- DATA[, same_vars] %>% group_by(SEX) %>%
  dplyr::select(all_of(same_vars.DiscreteDistribution)) %>% summarise_all(mean, na.rm = T)

DATA_summary_mean <- DATA[, same_vars] %>% group_by(SEX) %>%
  dplyr::select(all_of(same_vars.ContinouseDistribution)) %>% summarise_all(mean, na.rm = T)

DATA_summary_sd <- DATA[, same_vars] %>% group_by(SEX) %>%
  dplyr::select(all_of(same_vars.ContinouseDistribution)) %>% summarise_all(sd, na.rm = T)

DATA = SimulateData_GammaDist(SEQN = imputed_data$SEQN,
                               proportion_summ = nhanes_summary_prop,
                               mean_summ = nhanes_summary_mean,
                               sd_summ = nhanes_summary_sd,
                               strata_var = "SEX",
                               strata_prop = 0.5)

same_vars <- c("SEX", "AGE", "TOTCHOL", "SYSBP", "DIABP", "CURSMOKE", "DIABETES", "BPMEDS",
              "HDL", "BMI")
DATA_summary_prop1 <- DATA[, same_vars] %>%
  group_by(SEX) %>%
  dplyr::select(all_of(same_vars.DiscreteDistribution)) %>%
  summarise_all(mean, na.rm = T)

DATA_summary_mean1 <- DATA[, same_vars] %>%
  group_by(SEX) %>%
  dplyr::select(all_of(same_vars.ContinouseDistribution)) %>%
  summarise_all(mean, na.rm = T)

DATA_summary_sd1 <- DATA[, same_vars] %>%
  group_by(SEX) %>%

```

```

dplyr::select(all_of(same_vars.ContinouseDistribution)) %>%
summarise_all(sd,na.rm = T)

# For display
kable(nhanes_summary_prop, caption = "Summary of NHANES Data Proportions")
kable(DATA_summary_prop, caption = "Summary of Simulated Data Proportions-Normal Dist")
kable(DATA_summary_prop1, caption = "Summary of Simulated Data Proportions-Gamma Dist")

kable(nhanes_summary_mean, caption = "Summary of NHANES Data Mean")
kable(DATA_summary_mean, caption = "Summary of Simulated Data Mean-Normal Dist")
kable(DATA_summary_mean1, caption = "Summary of Simulated Data Mean-Gamma Dist")

kable(nhanes_summary_sd, caption = "Summary of NHANES Data SD")
kable(DATA_summary_sd, caption = "Summary of Simulated Data SD-Normal Dist")
kable(DATA_summary_sd1, caption = "Summary of Simulated Data SD-Gamma Dist")

# table(DATA$CURSMOKE,imputed_data$CURSMOKE) %>% kable
# table(DATA$DIABETES,imputed_data$DIABETES) %>% kable
# table(DATA$BPMEDS,imputed_data$BPMEDS)%>% kable

DATA2 = DATA %>%
  dplyr::filter(AGE>= 30 & AGE <= 62)%>%
  round%>%
  dplyr::mutate(Simulated="YES")
imputed_data2 = imputed_data %>%
  dplyr::select(colnames(DATA)) %>%
  mutate(Simulated="NO")

ggplotData<-bind_rows(imputed_data2,DATA2)

plt1<-ggplotData %>%
  ggplot() +
  geom_histogram(aes(x=AGE,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()
plt2<-ggplotData %>%
  ggplot() +
  geom_histogram(aes(x=TOTCHOL,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()

plt3<-ggplotData %>%
  ggplot() +
  geom_histogram(aes(x=SYSBP,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()
plt4<-ggplotData %>%
  ggplot() +

```

```

    geom_histogram(aes(x=HDLc,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
    theme_bw()
plt5<-ggplotData %>%
  ggplot() +
  geom_histogram(aes(x=BMI,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()

(plt1|plt2)/(plt3|plt4)/(plt5)

DATA = SimulateData_GammaDist(SEQN = imputed_data$SEQN,
                              proportion_summ=nhanes_summary_prop,
                              mean_summ =nhanes_summary_mean ,
                              sd_summ =nhanes_summary_sd ,
                              strata_var = "SEX",
                              strata_prop =0.5)

DATA2<-DATA %>%
  dplyr::filter(AGE>= 30 & AGE <= 62)%>%
  round%>%
  dplyr::mutate(Simulated="YES")
imputed_data2 = imputed_data %>%
  dplyr::select(colnames(DATA)) %>%
  mutate(Simulated="NO")

ggplotData<- bind_rows(imputed_data2,DATA2)

plt1<-ggplotData %>% ggplot() +
  geom_histogram(aes(x=AGE,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()
plt2<- ggplotData %>% ggplot() +
  geom_histogram(aes(x=TOTCHOL,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()

plt3<- ggplotData %>% ggplot() +
  geom_histogram(aes(x=SYSBP,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()
plt4<-ggplotData %>% ggplot() +
  geom_histogram(aes(x=HDLc,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()
plt5<-ggplotData %>% ggplot() +
  geom_histogram(aes(x=BMI,color=Simulated,y=after_stat(density)),alpha=0.6,binwidth = 0.5)+
  theme_bw()

(plt1|plt2)/(plt3|plt4)/(plt5)

DATA = SimulateData_GammaDist(SEQN = imputed_data$SEQN,
                              proportion_summ=nhanes_summary_prop,
                              mean_summ =nhanes_summary_mean ,
                              sd_summ =nhanes_summary_sd ,
                              strata_var = "SEX",
                              strata_prop =0.5)

par(mfrow = c(1, 2), mar = c(2, 1, 1, 1))

```

```

DATA = DATA %>%
  as.data.frame()%>%
  dplyr::select(order(colnames(DATA))) %>%
  dplyr::select(-SEQN)
M = cor(DATA)
corrplot(M, method = 'number', tl.cex = 0.5)
imputed_data2 = imputed_data2[, colnames(DATA)]
M = cor(imputed_data2)
corrplot(M, method = 'number', tl.cex = 0.5)

N=1000

MC_brier_estimate = rep(0,N)

for (l in 1:N) {

DATA = SimulateData_NormalDist(SEQN = imputed_data$SEQN,
                                proportion_summ=nhanes_summary_prop,
                                mean_summ =nhanes_summary_mean ,
                                cov_matrix, # Add covariance matrix parameter
                                sd_summ =nhanes_summary_sd ,
                                strata_var = "SEX",
                                strata_prop =0.5)

MC_brier_estimate[l] = BrierEstimate(DATA = DATA)

}

hist(MC_brier_estimate,main="Overall Brier estimate distribution",xlab="Monte Carlo Brier estimates",fr
abline(v=brier_estimate,col="red",lty=2)
abline(v=mean(MC_brier_estimate), col="blue", lty=2)
abline(v=mean(test_brier), col="green", lty=2)
# Add legend
legend("topright", legend = c("NHANES Brier Estimate", "Mean MC Brier", "Mean Test Brier"),
      col = c("red", "blue", "green"), lty = 2)

N=1000

MC_brier_estimate = rep(0,N)
imputed_data_men<-imputed_data%>%filter(SEX==1)

for (l in 1:N) {

DATA = SimulateData_NormalDist(SEQN = imputed_men$SEQN,
                                proportion_summ=nhanes_summary_prop,
                                mean_summ =nhanes_summary_mean ,
                                cov_matrix, # Add covariance matrix parameter
                                sd_summ =nhanes_summary_sd)

```

```

MC_brier_estimate[l] = BrierEstimate(DATA = DATA)

}

hist(MC_brier_estimate,main="Brier estimate distribution Men",xlab="Monte Carlo Brier estimates",freq =
abline(v=brier_estimate_men,col="red",lty=2)
abline(v=mean(MC_brier_estimate), col="blue", lty=2)
abline(v=mean(test_brier_men), col="green", lty=2)
# Add legend
legend("topright", legend = c("NHANES Brier Estimate", "Mean MC Brier", "Mean Test Brier"),
      col = c("red", "blue", "green"), lty = 2)
N=1000

MC_brier_estimate = rep(0,N)
imputed_data_women<-imputed_data%>%filter(SEX==2)

for (l in 1:N) {

DATA = SimulateData_NormalDist(SEQN = imputed_women$SEQN,
                                proportion_summ=nhanes_summary_prop,
                                mean_summ =nhanes_summary_mean ,
                                cov_matrix, # Add covariance matrix parameter
                                sd_summ =nhanes_summary_sd)

MC_brier_estimate[l] = BrierEstimate(DATA = DATA)

}

hist(MC_brier_estimate,main="Brier estimate distribution Women",xlab="Monte Carlo Brier estimates",freq =
abline(v=brier_estimate_women,col="red",lty=2)
abline(v=mean(MC_brier_estimate), col="blue", lty=2)
abline(v=mean(test_brier_women), col="green", lty=2)
# Add legend
legend("topright", legend = c("NHANES Brier Estimate", "Mean MC Brier", "Mean Test Brier"),
      col = c("red", "blue", "green"), lty = 2)

```