PHP2550 Project 3: Simulation Studies

Evaluating the Performance of a Prediction Model in Different Population

Yiwen Liang

12/03/2023

Abstract

Background: Evaluating the transportability of prediction models becomes crucial when considering their application in diverse target populations. Various methods have emerged to assess model performance in different populations, and this project focuses on transporting a cardiovascular disease (CVD) events model developed using Framingham Heart Study data to the broader National Health and Nutrition Examination Survey (NHANES) data in 2017.

Methods: Three evaluations are conducted to gauge the model's performance in different scenarios. Firstly, in the original population of the Framingham study where the model is developed, Brier scores can be easily computed. Secondly, considering transporting the model to a new target population where individual-level data is accessible whereas the outcome is lacking, an alternative Brier score estimator proposed by Dr. Jon Steingrimsson is employed to assess the model's performance. Lastly, when only summary statistics of the target population are available, a simulated target population is created based on these statistics, and the estimated Brier score is utilized to assess the model's performance in the simulated population. The Brier scores and estimates from these evaluations are scrutinized to ascertain the model's accuracy and transportability.

Results: The Brier score is 0.1992 for men and 0.1118 for women within Framingham, and the averaging estiamted Brier score is 0.1323 for men and 0.0670 for women in NHANES. The Brier score from the simulation study have the mean of 0.1439 for men and the mean of 0.0573 with the standard deviation for women, which are close to the estiamted Brier scores using the origianl target population with individual-level data.

Conclusions: Results indicate that the prediction model exhibits reasonable accuracy for CVD risk prediction, with better performance in females. Strong performance in simulation studies indicates substantial transportability of the cardiovascular disease events model from Framingham Heart Study data to the extensive NHANES data. Additionally, if the simulation of the target population is appropriately conducted, it's applicable for scientists to apply transportability analysis to a simulated data when the full data is not available.

1. Introduction

The development of a prediction model typically originates from the goal of predicting outcomes in a target population. In certain scenarios, it's feasible to construct the model using the target population or the subset of the population where the outcome variable is available. The mean squared error (MSE) (or Brier score for binary outcomes) is a valuable metric for evaluating the predictive accuracy of the model in predicting the outcome of interest or assessing the transportability of the prediction model to the actual target population. However, in most practical situations, it's more common to lack outcome data in a target population. Consequently, the prediction of outcomes relies on a series of other available covariates. In the absence of outcome data in the target population, the Brier score becomes impractical.

This project endeavors to find out whether transportability analysis can be applied to simulated data when full data is not available, by evaluating the performance of the CVD prediction model developed from the population in the Framingham Heart Study, in the population from National Health and Nutrition Examination Survey (NHANES) in 2017, and in the simulated target population, thereby offering valuable insights into the generalizability of this model.

The project includes three key evaluations. The first is to compute the Brier scores for the prediction model within the population for which it was originally developed. The second is to estimate the Brier score of the model in a distinct target population using the estimator proposed by Dr. Jon Steingrimsson, which allows the assessment of the model's performance in a new population. In this scenario, the individual-level covariates are accessible. The last evaluation is similar to the second one, but in this case, we'll simulate a target population using only summary statistics when individual-level data is not available. By comparing the Brier scores and estimates derived from these three evaluations, the project seeks to ascertain the feasibility of applying transportability analysis to simulated data in instances where complete individual-level data is unavailable.

The data sources used in this project include data from `Framingham` and `NHANES` in 2017, and both are publicly available (in package `riskCommunicator` (Grembi 2022) and `nhanesA` (Endres 2023)). The code files provided and used in this project, along with three `RData` files with simulation results stored are available at Github, https://github.com/yiwen-liang/PHP_2550_Final_Portfolio.

2. Methods

2.1 Data

In this project, two datasets are employed for distinct purposes. The `Framingham` data serves as the foundation for constructing predictive models aimed at assessing cardiovascular risks. The variable selection and model fitting procedure are designed to replicate the models outlined in *General Cardiovascular Risk Profile for Use in Primary Care* (D'Agostino RB Sr 2008). On the other hand, the `NHANES` data is the target population against which the performance of the constructed models is evaluated. Additionally, we also intend to conduct the same transportability analysis on the simulates target population based on the summary statistics of the `NHANES` data at a later stage in this project.

2.1.1 Framingham

The `framingham` dataset used in this project is derived from the Framingham Heart Study, a comprehensive, long-term prospective investigation conducted in Framingham, Massachusetts, and the data is available in the `riskCommunicator` package. Initiated in 1948, the study's primary objectives is to ascertain the underlying causes of cardiovascular disease (CVD). The initial cohort comprised 5209 subjects, and since its inception, participants have undergone biannual examinations, coupled with regular monitoring for cardiovascular outcomes. The dataset includes valuable information on common risk factors and disease markers, such as blood pressure, smoking history, medication use, all of which are documented in clinic examination data. Criteria for defining CVD, along with detailed descriptions of its design and procedures have been well reported. A crucial eligibility criterion for inclusion in the `Framingham` study, and one directly pertinent to our analyses, is that participants must have been between 30 and 74 years old at the time fo their initial enrollment (D'Agostino RB Sr 2008).

2.1.2 NHANES

The "target" population of this project is the data in 2017 from the *National Health and Nutrition Examination Survey* (`NHANES`). `NHANES` has been an ongoing study since 1999, and for the purpose of this project, we utilized the `nhanesA` package to extract relevant data. This package is specifically designed for retrieving data from `NHANES` and is instrumental in our analyses. Our focus is on a subset of variables included in the models, rather than considering all variables available in the broader `Framingham` study. This streamlined approach ensures that our analyses and simulation center around the key variables pertinent to our research objectives.

In the last scenario of this project, where individual-level covariates are unavailable, we assume that only the summary statistics of `NHANES` are available. The simulation study will be conducted using these statistics of the target population.

2.2 Models

The model for the evaluation can be expressed as follows:

$$\log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 \log(\text{HDLC}) + \beta_2 \log(\text{TOTCHOL}) + \beta_3 \log(\text{AGE}) + \beta_4 \log(\text{SYSBP\_UT+1}) + \beta_5 \log(\text{SYSBP\_T+1}) + \beta_6 \text{CURSMOKE} + \beta_7 \text{DIABETES},$$

where $\text{E}(Y) = P(Y = 1) = \pi$. Here, $Y = 1$ denotes the occurrence of CVD, while $Y = 0$ indicates the absence of CVD. In this logistic regression model, a logit link function is chosen. The model is fitted separately to male and female participants, resulting in two models, one for men and one for women. Both models share the same covariate vectors, as indicated in the equation. The use of distinct models for both genders enables a more nuanced understanding of the influences of risk factors on CVD risks for each gender.

As previously mentioned during the introduction of the datasets, our focus is specifically on the variables incorporated into the model. Theses variables are `HDLC` (HDL cholesterol), `TOTCHOL` (total cholesterol), `AGE`, `SYSBP_UT`, `SYSBP_T`, `CURSMOKE`, `DIABETES`, and `SEX` (for stratification). Of these, `SYSBP_UT` and `SYSBP_T` are newly generated based on the information from `BPMEDS` (indicating whether the individual is on anti-hypertensive medication) and `SYSBP` (representing systolic blood pressure in mmHg).

2.3 Estimator for Brier score in the target population

Given the binary outcome, CVD, the Brier score is considered to be the appropriate measure for assessing the accuracy of the probabilistic predictions generated by the models. This project evaluates the models in three different population:

1. `Framingham` dataset using train-test split.
2. Subset of `NHANES` data that meets the eligibility criteria of the `Framingham` study.
3. Simulated `NHANES` population, where only summary statistics are available.

In each analysis, the dataset will be subsetted by the age criteria (30-74), a specification that was employed during the development of the model (D'Agostino RB Sr 2008). This age range is consistent with the parameters used in the original model construction and ensures that the analyses are confined to the relevant age group. We'll then conduct the transportability analyses under different scenarios to gain insights into the robustness and generalizability of the model. The primary challenge is that the outcome variable `CVD` is available only in the `Framingham` study, not in the `NHANES`. Consequently, we need a Brier score estimator that doesn't rely on the outcome data. We have opted to use the weighting estimator developed by Dr. Jon Steingrimsson (Steingrimsson JA 2023) to assess the model's performance in the latter two scenarios.

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

where S is the population indicator ($S = 1$ if from `Framingham`, $S = 0$ if from NHANES), D is the test-train indicator ($D = 0$ if in training set, $D = 1$ if in testing set), and $\hat{o}(X)$ is the estimator for the inverse-odds weights, $\frac{P(S=0|X,D_{test}=1)}{P(S=1|X,D_{test}=1)}$. $\frac{P(S=0|X,D_{test}=1)}{P(S=1|X,D_{test}=1)}$ can be obtained through modelling the probability of being in the source dataset, $P(S = 1)$, conditioning on all covariates in the predictive model and train-test indicator D. The computation is as follows:

$$\log(\frac{P(S=1|X,D_{test}=1)}{P(S=0|X,D_{test}=1)}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \beta_{p+1} D$$
$$\frac{P(S=1|X,D_{test}=1)}{P(S=0|X,D_{test}=1)} = \exp(\beta_0 + \beta_1 x_1 + ... + \beta_p x_p) + \beta_{p+1} D$$
$$\frac{P(S=0|X,D_{test}=1)}{P(S=1|X,D_{test}=1)} = \frac{1}{\exp(\beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \beta_{p+1} D)}$$

2.4 Monte Carlo Simulation

The simulation is planned and will be reported in the ADEMP structure (Morris TP 2019).

- **Aims:** The aim of this simulation is to assess the transportability of a prediction model initially developed using the Framingham dataset for Cardiovascular Disease (CVD) events, for the use in a different target population where only summary statistics are available. The transportability will be evaluated using the Brier score.
- **Data-generating mechanisms:** Data are simulated for male and female separately on $n_{men} = 4557$ and $n_{women} = 4697$ (number of male and female participants in `NHANES`). Here, we assume that individual-level information of the target population is unavailable. However, given the distribution of covariates in `Framingham` dataset, we may determine the theoretical distribution for each of the seven variables based on the summary statistics of NHANES and the distributions observed in the `Framingham` data. Subsequently, we identify the subset of the simulated population that meets the eligibility criteria of the `Framingham` study (individuals between 30 and 74 years old).

We present out two attempts of simulating these covariates in the target population. One is to simply simulate each variable by its own summary statistics and its distribution in the source population, where we assume that these variables are independent of each other. Due to the distinctive distribution characteristics of `AGE`, we opt to simulate it using three different distributions: 1) **Uniform(1,75)**, 2) **Beta(10,19)**, and 3) **truncated Normal distribution**, with lower bound = 1, and upper bound = 75. The distributions and specified parameters for the other six variables, assuming independence, are presented in Table 1. How we determine them and the dimension of the dataset will be illustrated in the **Results** section **3.4.1**.

Table 1: Distributions for Simulation, Assuming Independence

Gender	HDLC	TOTCHOL	BPMEDS	SYSBP	CURSMOKE	DIABETES
	Gamma	Gamma	Binomial	Gamma	Binomial	Binomial
Men	(13.42, 0.27)	(19.14, 0.11)	(1, 0.28)	(42.86, 0.35)	(1, 0.20)	(1, 0.13)
Women	(14.56, 0.25)	(20.31, 0.11)	(1, 0.28)	(32.48, 0.27)	(1, 0.12)	(1, 0.06)

The second approach explores associations between variables. Through exploratory data analysis, it is observed that the distributions of continuous variables (excluding `AGE`) exhibit right skewness, which is mitigated by log-transformation, resulting in approximately normal distributions. Considering the associations between variables, a choice is made in favor of a multivariate normal distribution. Additionally, three categorical (binary) variables are found to be correlated with the continuous variable. To maintain these correlations, the simulation for categorical variables will also be drawn from the multivariate normal distribution. Subsequently, the continuous results will be converted back to binary categories based on quantiles for the three binary variables, thus preserving the underlying associations.

- **Estimands:** The weighting estimator developed by Dr. Jon Steingrimsson (Steingrimsson JA 2023) to assess the model's performance in the latter two scenarios.

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

where S is the population indicator ($S = 1$ if from `Framingham`, $S = 0$ if from NHANES), D is the test-train indicator ($D = 0$ if in training set, $D = 1$ if in testing set), and $\hat{o}(X)$ is the estimator for the inverse-odds weights, $\frac{P(S=0|X,D_{test}=1)}{P(S=1|X,D_{test}=1)}$. More details regarding computation are described in **2.3**.

- **Methods:** For each simulated dataset, we first apply the eligibility criteria (30-74 years old) before combining it with the `Framingham` dataset, and subsequently compute the estimated Brier score.
- **Performance measures:** In order to assess the transportability of the CVD prediction model when only summary statistics of the target population is available, we would compare the estimated Brier scores from the simulation to the Brier score estimates from NHANES population, where individual-level covariates are accessible. The metrics employed include convergence and empirical standard errors.

3. Results

3.1 Data Summary

The summaries of the source population `Framingham` and the target population `NHANES`, both stratified by gender, are provided in Table 2. `Framingham` data is used for model development, and as shown in Table 2, we can see that our source population differs from our target population `NHANES` in several variables, smoking status (`CURSMOKE`) and whether in use of any anti-hypertensive dedication (`BPMEDS`), for example. This finding further demonstrates the need for transportability analysis.

Table 2: Summary of the Variables in Framingham and NHANES

	Framingham			NHANES		
SEX	1, N = 1,094	2, N = 1,445	p-value	1, N = 4,557	2, N = 4,697	p-value
CVD	360 (33%)	242 (17%)	<0.001			
TOTCHOL	226.44 (41.49)	246.32 (45.51)	<0.001	176.68 (40.38)	182.94 (40.59)	<0.001
AGE	60.01 (8.18)	60.55 (8.40)	0.13	34.12 (25.75)	34.55 (25.25)	0.3
SYSBP	138.94 (20.89)	139.94 (23.71)	0.6	122.49 (18.71)	120.20 (21.09)	<0.001
CURSMOKE	425 (39%)	445 (31%)	<0.001	596 (21%)	425 (14%)	<0.001
DIABETES	96 (8.8%)	95 (6.6%)	0.037	484 (11%)	409 (9.0%)	0.001
BPMEDS	123 (11%)	259 (18%)	<0.001	797 (28%)	853 (28%)	>0.9
HDLC	43.63 (13.37)	53.07 (15.67)	<0.001	49.57 (13.53)	57.01 (14.94)	<0.001
BMI	26.25 (3.47)	25.55 (4.22)	<0.001	26.16 (7.63)	26.98 (8.80)	0.014
SYSBP_UT	121.04 (46.69)	111.49 (55.89)	<0.001	84.09 (58.42)	79.70 (56.44)	<0.001
SYSBP_T	17.90 (50.93)	28.45 (61.53)	<0.001	33.23 (58.68)	33.80 (60.44)	0.8

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

[2] Pearson's Chi-squared test; Wilcoxon rank sum test

[3] Mean (SD); n (%)

[4] Wilcoxon rank sum test; Pearson's Chi-squared test

3.2 Performance Evaluation in Framingham

We begin by evaluating the model's performance within the `Framingham` dataset. Given the availability of the outcome variable, we split the `Framingham` into "train" and "test", allocating 70% for training and 30% for testing, and the Brier scores are then easily computed, as presented in Table 3.

Table 3: Brier Scores of the Model in the Framingham Dataset

Men	Women
0.1993	0.1118

3.3 Performance Evaluation in the Target Population (NHANES)

We proceed to evaluate the model's performance in the target population underlying NHANES. In order to obtain the Brier score estimates in the target population, the following steps are taken: 1) identifying the subset of `NHANES` eligible for the `Framingham` study, 2) combining it with `Framingham`, 3) creating S and D variables as defined in Equation (1), and 4) computing the estimated Brier risk in the target population.

The `NHANES` data in 2017 has 9254 records originally, and 4060 of them are from individuals between 30 and 74 years old (1961 men and 2099 women).

3.3.1 Missing Pattern in NHANES

First, we conduct an examination of missing data in `NHANES`. As shown in Table 5, we observe that the missingness is not severe. `BPMEDS` and `CURSMOKE` exhibit the highest proportion of missing data, with greater than 37% for men and 35% for women. Besides, we notice a similar missing pattern for `HDLC` and `TOTCHOL`. Given that both variables are associated with cholesterol, it's likely that the information on these two variables was obtained through the same examination, leading to them missing together.

Table 4: Summary of Missing Values in NHANES

Variables	Men		Women	
	N	Proportion (%)	N	Proportion (%)
BMI	660	14.48	589	12.54
BPMEDS	1722	37.79	1676	35.68
CURSMOKE	1717	37.68	1681	35.79
DIABETES	191	4.19	170	3.62
HDLC	1280	28.09	1236	26.31
SYSBP	1442	31.64	1510	32.15
SYSBP_T	1849	40.57	1830	38.96
SYSBP_UT	2019	44.31	2029	43.20
TOTCHOL	1280	28.09	1236	26.31

We have two approaches to handle missing data: one involves removing all records with missing information, and the other entails imputing values through multiple imputation.

1. Drop All Missing Records: This would reduce the number of observations from 4060 to 3001, resulting in the removal of approximately 25% of the records. While this method eliminates missingness, it comes at the cost of losing some information.

Table 5: Estimated Brier Scores of the Model in the NHANES Dataset (Drop NA)

Men	Women
0.142278	0.074679

2. Multiple Imputation (MI): We use `mice()` function to perform multiple imputation, generating 5 imputed datasets with the seed set to 2550 to offset the random number generator. Table 7 presents the Brier score estimates for each imputed dataset, with the averaging Brier score estimates (highlighted in bold in the last row) provided both for males and females.

Table 6: Estimated Brier Scores of the Model in the NHANES Dataset (MI)

Men	Women
0.131938	0.067172
0.133760	0.067358
0.132118	0.066761
0.132421	0.065936
0.131182	0.067645
0.132284	**0.066974**

3.4 Performance Evaluation in the Simulated Target Population (NHANES)

In this section, we assume that individual-level data is not available from the target population (`NHANES`), and we'll simulate the individual level data based solely on the summary statistics.

3.4.1 Attempt 1: Simulation Using Individual Variables

Table 2 provides the number of participants, mean, and standard deviation of each variable by gender. We plan to simulate 4557 observations for men and 4697 observation for women, with each record comprising the seven variables included in the model. Additionally, for a more accurate simulation, we will reference the distribution of each variable in the original population (`Framingham`) used for constructing models.

Among the seven variables of interest, four of them are continuous, and their distributions in the `Framingham` data are in Figure 1. We can see that `HDLC`, `TOTCHOL`, and `SYSBP` all have right-skewed distributions, and considering the range of these variables, we decide to use Gamma distribution for simulation. We know that the mean and the standard deviation of the Gamma distribution can be expressed as $\mu = \frac{\alpha}{\beta}$, $\sigma = \frac{\sqrt{\alpha}}{\beta}$, where α is the shape and β is the rate parameter. By plugging in the summary statistics of `NHANES`, we solve for the parameters of the Gamma distribution in Table 1. The binomial distribution is employed to simulate three binary variables, and `n` and `p` parameters are obtained similarly.

The primary challenge lies in the variable `AGE`. The means and standard deviations of age for males and females are 34.12 (25.75) and 34.55 (25.25), respectively. However, the distribution of `AGE` in `Framingham` ranges roughly between 45 and 80, exhibiting multiple noticeable spikes. It's impractical to generate a vector of age values that mirrors the distribution of `AGE` in `Framingham` while maintaining the same summary statistics observed in `NHANES`. Overall, I've selected the following three distributions to explore:

1. Uniform(1,75): Assuming that we don't have any prior knowledge on the distribution, in order to have a similar mean and standard deviation, I opt for the uniform distribution and set the bounds to $[1, 75]$.
2. Beta(10,19)*100: Since age should be strictly positive and the distribution of `AGE` in `Framingham` is right-skewed, I attempt to simulate the age with Beta distribution (values range between 0 and 1), then multiply them by 100 as age.
3. Truncated Normal Distribution: Normal is the continuous distribution that usually comes first to mind. But given the mean and standard deviation (34.12 (25.75) and 34.55 (25.25)), it's inevitable to have negative values. Hence I try using the truncated normal distribution with lower bound = 1, and upper bound = 75, and the mean and standard deviation are assigned exactly as the summary statistics.

The three continuous variables are reasonable. However, there's no significant difference in the performance among the three distributions for `AGE`. So we'll simulate datasets using all three approaches for `AGE`. In

each iteration, the simulated dataset will be subsetted by the age criteria (30-74) of the `Framingham` study, combined with `Framingham` data, and the estimated Brier score in the simulated target population for men and women will be computed, respectively.

Through a small number of simulations, we have that the variance of the estimated brier scores are usually less than 1e-4. Given that

$$\text{Monte Carlo SE(Bias)} = \sqrt{\text{Var}/n_{sim}},$$

it's easy to achieve a Monte Carlo SE of Bias lower than, for example, 0.001. Therefore, we set the number of repetitions to 1000, a commonly used number for Monte Carlo simulations, without further detailed computation.

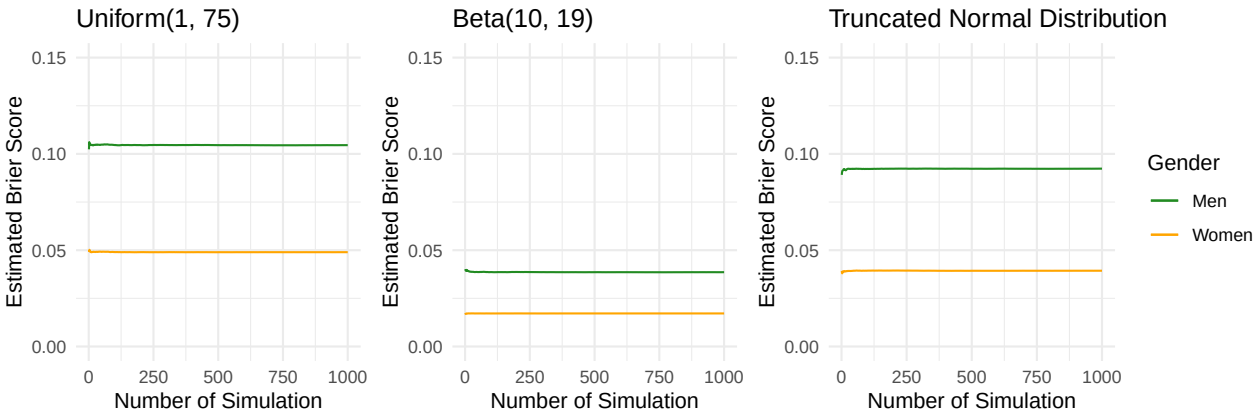

Figure 1: Cumualtive Average of the Estimates of Brier Scores for Three Simulations

Table 7: Average Estimated Brier Scores of the Model in Simulated Population, Three Independent Simulations

		Simulated NHANES (Uniform)	Simulated NHANES (Beta)	Simulated NHANES (Truncated Normal)
Men	Mean	0.1045	0.0386	0.0923
	SE	0.0020	0.0012	0.0019
Women	Mean	0.0490	0.0172	0.0393
	SE	0.0012	0.0003	0.0010

Figure 1 shows the cumulative average of estimated Brier scores. Leveraging the law of large numbers, the Monte Carlo simulation employs random sampling to produce multiple simulated results, and their mean is expected to converge towards the true value. We can see that the empirical means in all three simulation settings converge well. In Table 7, we present the averaging estimated Brier scores and their empirical standard errors in the simulated target population `NHANES` with 3 different distributions for `AGE`.

3.4.2 Attempt 2: Simulation Involving Association Between Variables

Here, we revise the data-generating mechanism used in **3.4.1**, which previously overlooked the association between the covariates. We treat both continuous and categorical variables as continuous, drawing samples from a single multivariate normal distribution. Based on our exploratory analysis, continuous variables will be simulated on a log scale, exhibiting a normal distribution. The continuous results for categorical

variables will then be converted back to binary categories based on quantiles. By specifying the means and covariance matrix of all variables, this simulation effectively generates reasonable values for each variable while preserving the inherent associations among them.

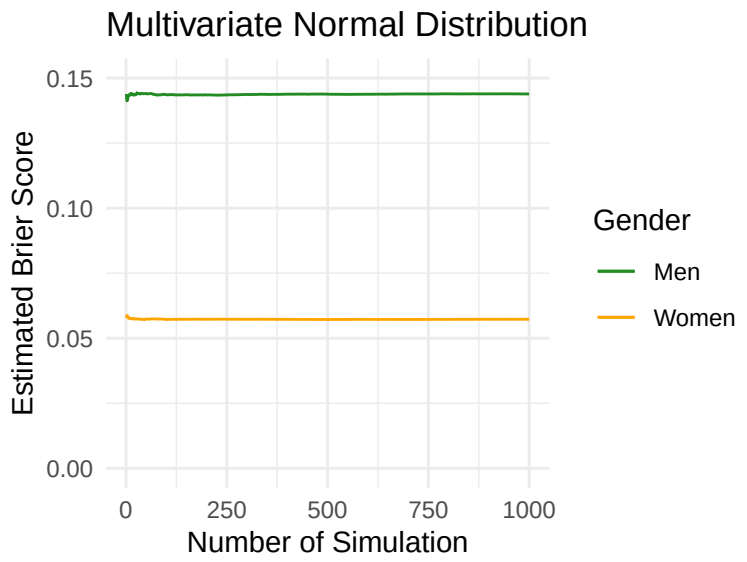

Figure 2: Cumualtive Average of the Estimates of Brier Scores, Multivariate Normal

Table 8: Average Estimated Brier Scores of the Model in Simulated Population, Multivariate Normal

		Simulated NHANES (Multivariate Normal)
Men	Mean	0.1439
	SE	0.0034
Women	Mean	0.0573
	SE	0.0014

Figure 2 illustrates the cumulative average of the estimated Brier score when employing a multivariate normal distribution for simulation. It is evident that the empirical mean converges effectively in this scenario. Table 8 provides the average estimated Brier score along with its empirical standard error in the simulated target population using a multivariate normal distribution. For ease of comparison, Table 9 is created, presenting Brier scores observed within the `Framingham` data, estimated Brier scores in the target population `NHANES` (individual-level data available) after excluding all missing entries and conducting multiple imputations (averaging estimates), and averaging estimated Brier scores in the simulated target population with four different simulation settings.

Table 9: Brier Scores and Average Estimated Brier Scores of the Model in Different Population

	Framingham	NHANES (Drop NA)	NHANES (MI)	Simulated NHANES (Uniform)	Simulated NHANES (Beta)	Simulated NHANES (Truncated Normal)	Simulated NHANES (Multivariate Normal)
Men	0.1993	0.1423	0.1323	0.1045	0.0386	0.0923	0.1439
Women	0.1118	0.0747	0.0670	0.0490	0.0172	0.0393	0.0573

In Table 9, we observe that among the four simulation settings we tested, the estimated Brier score from the one adopting the multivariate normal distribution (0.1439 for men and 0.0573 for women) is the closest to the

Brier score estimate obtained from the original NHANES data (0.1323 for men and 0.0670 for women). This indicates that the simulation using the multivariate normal distribution may be the most accurate among the four. This aligns with our expectations, as assuming independence among health-related variables would lead to a loss of information. For instance, whether the subject is on blood pressure medication is evidently associated with their systolic blood pressure. Maintaining these correlations in the simulation results in a better representation. Therefore, for further discussion in the next section, we will place more emphasis on the simulation with multivariate normal distribution.

4. Discussion

The Brier score of a model is commonly used to assess the model's predicted probabilities against observed probabilities. The Brier score always falls between 0 and 1, and the closer the score is to 0, the better the predictive accuracy. Generally, based on Table 9 in the **Results** section, it's evident that the Brier scores and estimates are higher for males than for females in all scenarios. In other words, the prediction model demonstrates better accuracy in predicting CVD risk for women than for men.

In essence, the model plays a crucial role in predicting the risk of cardiovascular disease, enabling clinicians to take preventive measures or mitigate risks by addressing these risk factors early on. For the estimator of Brier scores, it allows scientists to assess the model's performance when transported to another target population when outcome data is unavailable, thereby carrying substantial practical implications.

Moreover, as shown in Table 9, the Brier scores for the provided model, when the Framingham data is split into training and testing sets, are 0.1993 for males and 0.1118 for females. The Brier scores estimated within NHANES are 0.1323 for men and 0.0670 for women after multiple imputation, considering these as the baseline for comparison. Surprisingly, the estimated Brier scores in the target population NHANES are smaller than the Brier risks obtained within the Framingham data, against which the models are constructed. Given the differences in demographic characteristics between the two populations, it is unexpected that the predictive accuracy of the model increases when being transported to a new population.

Regarding simulated populations, we observe that the estimated Brier scores in three simulated target population, assuming independent variables, are smaller than the estimates from the non-simulated population and even smaller than those within the Framingham data. One possible explanation for this phenomenon is that our simulation references the distribution of variables in the Framingham, making it more similar to the original data used for building models, compared to the actual NHANES dataset. Additionally, since we opt for theoretical distributions (Gamma, binomial, uniform, etc.) for simulations, the simulated population is likely to exhibit less variability than the real population, in comparison to both Framingham and NHANES. Notably, there's no outliers in any of our simulated populations. Both factors contribute to result that the estimates of Brier risk in the NHANES are the greatest, while the estimates in the simulated target population are the smallest. On the other hand, after considering associations between variables, the simulation becomes more similar to the original NAHNES dataset, with close Brier score estimates.

The results imply that this CVD risk prediction model demonstrates reasonable accuracy in predicting CVD, particularly exhibiting higher accuracy for females. Furthermore, the evaluation of the model's performance in both a new target population and a simulated target population establishes its outstanding transportability. However, the observed decrease in Brier score estimates falls outside our expectations. In conclusion, the results suggest that, if the simulation of the target population is conducted appropriately, it is feasible for scientists to apply transportability analysis to a simulated data when the full data is not available.

The main limitation of this project is that we only estimated the Brier scores in the transportability analyses. The area under the ROC curve (AUC) is another commonly used measure for assessing transportability, and Bing Li's paper (Li B 2023) proposes an estimator for AUC when outcome data is unavailable.

5. Conclusion

Our project suggests that this CVD risk prediction model demonstrates a relatively high accuracy in predicting CVD, with a higher accuracy for females in particular. Its exceptional transportability is established by the model's performance evaluation in both a simulated and a new target population. Our findings illustrate that if the simulation of the target population is appropriately conducted, it's applicable for scientists to apply transportability analysis to a simulated data when the full data is not available.

References

D'Agostino RB Sr, Pencina MJ, Vasan RS. 2008. "General Cardiovascular Risk Profile for Use in Primary Care." *Circulation* 117 (6): 743–53. https://doi.org/10.1161/CIRCULATIONAHA.107.699579.

Endres, Christopher. 2023. *nhanesA: NHANES Data Retrieval*. https://CRAN.R-project.org/package=nhanesA.

Grembi, Jessica. 2022. *riskCommunicator: G-Computation to Estimate Interpretable Epidemiological Effects*. https://CRAN.R-project.org/package=riskCommunicator.

Li B, Dahabreh IJ, Gatsonis C. 2023. "Estimating the Area Under the ROC Curve When Transporting a Prediction Model to a Target Population." *Biometrics* 79 (3): 2382–93. https://doi.org/10.1111/biom.13796.

Morris TP, Crowther MJ, White IR. 2019. "Using Simulation Studies to Evaluate Statistical Methods." *Stat Med* 38 (11): 2074–2102. https://doi.org/10.1002/sim.8086.

Steingrimsson JA, Li B, Gatsonis C. 2023. "Transporting a Prediction Model for Use in a New Target Population." *Am J Epidemiol* 192 (2): 296–304. https://doi.org/10.1093/aje/kwac128.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)

# Load the packages
library(cowplot)
library(ggplot2)
library(ggpubr)
library(gt)
library(gtsummary)
library(kableExtra)
library(knitr)
library(MASS)
library(mice)
library(nhanesA)
library(reshape2)
library(riskCommunicator)
library(tableone)
library(tidyverse)
library(truncnorm)

# kable of parameters
sim_dist <- data.frame(HDLC = c("(13.42, 0.27)", "(14.56, 0.25)"),
           TOTCHOL = c("(19.14, 0.11)", "(20.31, 0.11)"),
           BPMEDS = c("(1, 0.28)", "(1, 0.28)"),
           SYSBP = c("(42.86, 0.35)", "(32.48, 0.27)"),
           CURSMOKE = c("(1, 0.20)", "(1, 0.12)"),
           DIABETES = c("(1, 0.13)", "(1, 0.06)"))
rownames(sim_dist) <- c("Men", "Women")

sim_dist %>%
  kbl(caption = "Distributions for Simulation, Assuming Independence",
      col.names = linebreak(c("Gamma", "Gamma", "Binomial",
                    "Gamma", "Binomial", "Binomial")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  add_header_above(c("Gender"=1, "HDLC"=1, "TOTCHOL"=1, "BPMEDS"=1,
                     "SYSBP"=1, "CURSMOKE"=1, "DIABETES"=1)) %>%
  kable_styling(full_width = FALSE,
                latex_options = c('HOLD_position'))

# Pre-processing Framingham dataset (provided)
data("framingham")

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

# 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)
```

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

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

# Save original version of Framingham for EDA later
df_eda <- framingham_df

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

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

```
  full_join(tchol_2017, by = "SEQN") %>%
  full_join(diq_2017, by = "SEQN")

df_2017$SYSBP_UT <- ifelse(df_2017$BPMEDS == 0, df_2017$SYSBP, 0)
df_2017$SYSBP_T <- ifelse(df_2017$BPMEDS == 1, df_2017$SYSBP, 0)

# Summary table for Framingham and NHANES
tbl_framingham <- tbl_summary(framingham_df, include = -c(DIABP),
                              by = SEX, missing = "no",
                              statistic = list(all_continuous() ~ "{mean} ({sd})"),
                              digits = all_continuous() ~ 2) %>%
  add_p() %>%
  modify_header(label = "SEX") %>%
  bold_labels()

tbl_nhanes <- tbl_summary(df_2017, include = -c(SEQN),
                          by = SEX, missing = "no",
                          statistic = list(all_continuous() ~ "{mean} ({sd})"),
                          digits = all_continuous() ~ 2) %>%
  add_p() %>%
  modify_header(label = "SEX") %>%
  bold_labels()

tbl_merge(list(tbl_framingham, tbl_nhanes),
          tab_spanner = c("Framingham", "NHANES")) %>%
  as_kable_extra(caption = "Summary of the Variables in Framingham and NHANES",
                 booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('HOLD_position', "striped"),
                font_size = 9)

# Split test and training datasets
# 70% of the data is used to construct the model
# 30% of the data is used to test the performance of the model
set.seed(2550)
samp_men <- sample(c(TRUE, FALSE), nrow(framingham_df_men),
                   replace=TRUE, prob=c(0.7,0.3))
train_men <- framingham_df_men[samp_men, ]
test_men <- framingham_df_men[!samp_men, ]

set.seed(2550)
samp_women <- sample(c(TRUE, FALSE), nrow(framingham_df_women),
                   replace=TRUE, prob=c(0.7,0.3))
train_women <- framingham_df_women[samp_women, ]
test_women <- framingham_df_women[!samp_women, ]

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

mod_women_tr <- glm(CVD ~ log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT+1) +
                      log(SYSBP_T+1) + CURSMOKE + DIABETES,
```

```
                    data = train_women, family = "binomial")

# Brier scores within Framingham
pred_men <- predict(mod_men_tr, newdata = test_men, type = "response")
pred_women <- predict(mod_women_tr, newdata = test_women, type = "response")

# Present Brier scores using kable
data.frame(Men = round(mean((pred_men-test_men$CVD)^2), 4),
           Women = round(mean((pred_women-test_women$CVD)^2), 4)) %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Brier Scores of the Model in the Framingham Dataset",
      col.names = linebreak(c("Men", "Women")),
      row.names = FALSE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('HOLD_position'))

# Assigned 70% D=0 in Framingham and get predictions
set.seed(2550)
framingham_df_men$D <- sample(c(1, 0), nrow(framingham_df_men),
                              replace=TRUE, prob=c(0.3,0.7))
framingham_df_women$D <- sample(c(1, 0), nrow(framingham_df_women),
                              replace=TRUE, prob=c(0.3,0.7))

mod_men <- glm(CVD ~ log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT+1) +
                 log(SYSBP_T+1) + CURSMOKE + DIABETES, family = "binomial",
               data = framingham_df_men[framingham_df_men$D==0,])

mod_women <- glm(CVD ~ log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT+1) +
                   log(SYSBP_T+1) + CURSMOKE + DIABETES, family = "binomial",
                 data = framingham_df_women[framingham_df_women$D==0,])

# Predictions
framingham_df_men$pred[framingham_df_men$D==1] <- predict(mod_men,framingham_df_men[framingham_df_men$D=
framingham_df_women$pred[framingham_df_women$D==1] <- predict(mod_women, framingham_df_women[framingham_

# Manipulate Framingham for later combining
framingham_df_men <- framingham_df_men %>%
  dplyr::select(c("CVD", "SEX", "HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
                  "SYSBP_T", "CURSMOKE", "DIABETES", "D", "pred")) %>%
  mutate(S = 1)
framingham_df_women <- framingham_df_women %>%
  dplyr::select(c("CVD", "SEX", "HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
                  "SYSBP_T", "CURSMOKE", "DIABETES", "D", "pred")) %>%
  mutate(S = 1)

# 26 out of 29 variables have missing value
descript1 <- df_2017 %>%
  summarise(
    N = colSums(is.na(df_2017[df_2017$SEX==1,])),
    prop = round(colMeans(is.na(df_2017[df_2017$SEX==1,]))*100, 2)) %>%
  mutate(Variables = colnames(df_2017)) %>%
  filter(N != 0) %>%
  as.data.frame()
```

```r
descript2 <- df_2017 %>%
  summarise(
    N = colSums(is.na(df_2017[df_2017$SEX==2,])),
    prop = round(colMeans(is.na(df_2017[df_2017$SEX==2,]))*100, 2)) %>%
  mutate(Variables = colnames(df_2017)) %>%
  filter(N != 0) %>%
  as.data.frame()

descript <- merge(descript1, descript2, by = "Variables", all.y = T)
descript[is.na(descript)] <- 0

# Display missing data summary table using kable
descript %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Summary of Missing Values in NHANES",
      col.names = linebreak(c("", "N", "Proportion (%)",
                              "N", "Proportion (%)")),
      row.names = FALSE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  add_header_above(c("Variables"=1, "Men"=2, "Women"=2)) %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'))

# BMI is not included in the model so we don't need to worry about it.
# For the other 6, 3 of them are binary variables
# CURSMOKE, BPMEDS, DIABETES

# Function: combine
comb_fun <- function(dt, sex="men") {
  dt$SYSBP_UT <- ifelse(dt$BPMEDS==0, dt$SYSBP, 0)
  dt$SYSBP_T <- ifelse(dt$BPMEDS==1, dt$SYSBP, 0)

  dt <- dt %>%
    filter(AGE>=30 & AGE<=74) %>%
    dplyr::select(c("SEX", "HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
                    "SYSBP_T", "CURSMOKE", "DIABETES")) %>%
    mutate(S = 0, D = 1, CVD = NA)

  if (sex=="men") {
    df_comb <- merge(dt, framingham_df_men, all = T)
  } else {
    df_comb <- merge(dt, framingham_df_women, all = T)
  }

  return(df_comb)
}

#nrow(df_2017)
#nrow(drop_na(df_2017))

df_2017_drop_na_men <- drop_na(df_2017) %>% filter(SEX == 1)
df_2017_drop_na_women <- drop_na(df_2017) %>% filter(SEX == 2)

set.seed(2550)
```

```r
df_drop_comb1_men <- comb_fun(df_2017_drop_na_men, sex="men")
df_drop_comb1_women <- comb_fun(df_2017_drop_na_women, sex="women")

df_2017 <- df_2017 %>%
  dplyr::select(c("SEX", "HDLC", "TOTCHOL", "AGE", "BPMEDS",
                  "SYSBP", "CURSMOKE", "DIABETES"))

df_2017_men <- df_2017 %>% filter(SEX == 1)
df_2017_women <- df_2017 %>% filter(SEX == 2)

# MI
imp.nhanes.men <- mice(df_2017_men, m=5, print=FALSE, seed=2550)
imp.nhanes.women <- mice(df_2017_women, m=5, print=FALSE, seed=2550)

# Combine Framingham and imputed NHANES
df_2017_men_imp <- vector("list", 5)
df_2017_women_imp <- vector("list", 5)
df_comb1_men <- vector("list", 5)
df_comb1_women <- vector("list", 5)

for (i in 1:5) {
  df_2017_men_imp[[i]] <- mice::complete(imp.nhanes.men, i)
  df_2017_women_imp[[i]] <- mice::complete(imp.nhanes.women, i)

  df_comb1_men[[i]] <- comb_fun(df_2017_men_imp[[i]], sex="men")
  df_comb1_women[[i]] <- comb_fun(df_2017_women_imp[[i]], sex="women")
}

# Function to estimate Brier score using inverse-odds weights
brier_est_fun <- function(dt) {
  m_o <- glm(S ~ log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT+1)
             + log(SYSBP_T+1) + CURSMOKE + DIABETES,
             family = "binomial", data = dt[dt$D==1,])

  dt$o_hat[dt$D==1] <- 1/predict(m_o, type="response")

  df_temp <- dt[dt$S==1 & dt$D==1,]

  sum(df_temp$o_hat*(df_temp$CVD-df_temp$pred)^2) / sum(dt$S==0 & dt$D==1)
}

# Estimated Brier score in NHANES (target population) for men and women
# Drop all missing records
brier_est1 <- data.frame(brier_est_fun(df_drop_comb1_men),
                         brier_est_fun(df_drop_comb1_women))

# Present Brier scores using kable
round(brier_est1, 6) %>%
  kbl(caption = "Estimated Brier Scores of the Model in the NHANES Dataset (Drop NA)",
      col.names = linebreak(c("Men", "Women")),
      row.names = FALSE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE, latex_options = c('HOLD_position'))
```

```r
# Estimated Brier score in NHANES (target population) for men and women
# Implementing multiple imputation
brier_est2 <- data.frame(matrix(NA, nrow=6, ncol=2))
colnames(brier_est2) <- c("Men", "Women")

for (i in 1:5) {
  brier_est2[i,1] <- brier_est_fun(df_comb1_men[[i]])
  brier_est2[i,2] <- brier_est_fun(df_comb1_women[[i]])
}

brier_est2[6,] <- c(mean(brier_est2[1:5,1]), mean(brier_est2[1:5,2]))

# Present Brier scores using kable
round(brier_est2, 6) %>%
  kbl(caption = "Estimated Brier Scores of the Model in the NHANES Dataset (MI)",
      col.names = linebreak(c("Men", "Women")),
      row.names = FALSE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position')) %>%
  row_spec(6, bold=TRUE) %>%
  row_spec(5, hline_after = TRUE)

# Function: simulation
# type %in% c("uniform", "beta", "normal") for AGE
sim_fun <- function(type="uniform") {
  n_sim_men <- 4557
  n_sim_women <- 4697

  sim_men <- data.frame(SEX = rep(1, n_sim_men))
  sim_women <- data.frame(SEX = rep(2, n_sim_women))

  # HDLC_men
  a = (49.57/13.53)^2
  b = a/49.57
  sim_men$HDLC <- rgamma(n_sim_men, shape=a, rate=b)

  # HDLC_women
  a = (57.01/14.94)^2
  b = a/57.01
  sim_women$HDLC <- rgamma(n_sim_women, shape=a, rate=b)

  # TOTCHOL_men
  a = (176.68/40.38)^2
  b = a/176.68
  sim_men$TOTCHOL <- rgamma(n_sim_men, shape=a, rate=b)

  # TOTCHOL_women
  a = (182.94/40.59)^2
  b = a/182.94
  sim_women$TOTCHOL <- rgamma(n_sim_women, shape=a, rate=b)

  # AGE #
  if (type=="uniform") {
```

```
    sim_men$AGE <- runif(n_sim_men, min=1, max=75)
    sim_women$AGE <- runif(n_sim_women, min=1, max=75)
  } else if(type=="beta") {
    sim_men$AGE <- 100*rbeta(n_sim_men, shape1=10, shape2=19)
    sim_women$AGE <- 100*rbeta(n_sim_women, shape1=10, shape2=19)
  } else if(type=="normal") {
    sim_men$AGE <- rtruncnorm(n=n_sim_men, a=1, b=75, mean=34.12, sd=25.75)
    sim_women$AGE <- rtruncnorm(n=n_sim_women, a=1, b=75, mean=34.55, sd=25.25)
  }

  # BPMEDS_men
  p = 1-(0.45^2/0.28)
  n = round(0.28/p)
  sim_men$BPMEDS <- rbinom(n_sim_men, n, p)

  # BPMEDS_women
  p = 1-(0.45^2/0.28)
  n = round(0.28/p)
  sim_women$BPMEDS <- rbinom(n_sim_women, n, p)

  # SYSBP_men
  a = (122.49/18.71)^2
  b = a/122.49
  sim_men$SYSBP <- rgamma(n_sim_men, shape=a, rate=b)

  # SYSBP_women
  a = (120.20/21.09)^2
  b = a/120.20
  sim_women$SYSBP <- rgamma(n_sim_women, shape=a, rate=b)

  # CURSMOKE_men
  p = 1-(0.41^2/0.21)
  n = round(0.21/p)
  sim_men$CURSMOKE <- rbinom(n_sim_men, n, p)

  # CURSMOKE_women
  p = 1-(0.35^2/0.14)
  n = round(0.14/p)
  sim_women$CURSMOKE <- rbinom(n_sim_women, n, p)

  # DIABETES_men
  p = 1-(0.31^2/0.11)
  n = round(0.11/p)
  sim_men$DIABETES <- rbinom(n_sim_men, n, p)

  # DIABETES_women
  p = 1-(0.29^2/0.09)
  n = round(0.09/p)
  sim_women$DIABETES <- rbinom(n_sim_women, n, p)

  return(list(sim_men, sim_women))
}
```

```
brier_est_mc <- vector("list", 3)
brier_est_mc[[1]] <- data.frame(matrix(ncol=4, nrow=0,
                                       dimnames=list(NULL, c("Men", "n_men",
                                                             "Women", "n_women"))))
brier_est_mc[[2]] <- data.frame(matrix(ncol=4, nrow=0,
                                       dimnames=list(NULL, c("Men", "n_men",
                                                             "Women", "n_women"))))
brier_est_mc[[3]] <- data.frame(matrix(ncol=4, nrow=0,
                                       dimnames=list(NULL, c("Men", "n_men",
                                                             "Women", "n_women"))))

for (i in 1:1000){
  set.seed(i+1)
  sim1_men <- sim_fun("uniform")[[1]]
  sim1_comb_men <- comb_fun(sim1_men[sim1_men$AGE>=30 &
                                       sim1_men$AGE<=74,], "men")
  brier_est_mc[[1]][i,1] <- brier_est_fun(sim1_comb_men)
  brier_est_mc[[1]][i,2] <- nrow(sim1_men[sim1_men$AGE>=30 & sim1_men$AGE<=74,])

  set.seed(i+2)
  sim1_women <- sim_fun("uniform")[[2]]
  sim1_comb_women <- comb_fun(sim1_women[sim1_women$AGE>=30 &
                                           sim1_women$AGE<=74,], "women")
  brier_est_mc[[1]][i,3] <- brier_est_fun(sim1_comb_women)
  brier_est_mc[[1]][i,4] <- nrow(sim1_women[sim1_women$AGE>=30 & sim1_women$AGE<=74,])

  set.seed(i+3)
  sim2_men <- sim_fun("beta")[[1]]
  sim2_comb_men <- comb_fun(sim2_men[sim2_men$AGE>=30 &
                                       sim2_men$AGE<=74,], "men")
  brier_est_mc[[2]][i,1] <- brier_est_fun(sim2_comb_men)
  brier_est_mc[[2]][i,2] <- nrow(sim2_men[sim2_men$AGE>=30 & sim2_men$AGE<=74,])

  set.seed(i+4)
  sim2_women <- sim_fun("beta")[[2]]
  sim2_comb_women <- comb_fun(sim2_women[sim2_women$AGE>=30 &
                                           sim2_women$AGE<=74,], "women")
  brier_est_mc[[2]][i,3] <- brier_est_fun(sim2_comb_women)
  brier_est_mc[[2]][i,4] <- nrow(sim2_women[sim2_women$AGE>=30 & sim2_women$AGE<=74,])

  set.seed(i+5)
  sim3_men <- sim_fun("normal")[[1]]
  sim3_comb_men <- comb_fun(sim3_men[sim3_men$AGE>=30 &
                                       sim3_men$AGE<=74,], "men")
  brier_est_mc[[3]][i,1] <- brier_est_fun(sim3_comb_men)
  brier_est_mc[[3]][i,2] <- nrow(sim3_men[sim3_men$AGE>=30 & sim3_men$AGE<=74,])

  set.seed(i+6)
  sim3_women <- sim_fun("normal")[[2]]
  sim3_comb_women <- comb_fun(sim3_women[sim3_women$AGE>=30 &
                                           sim3_women$AGE<=74,], "women")
  brier_est_mc[[3]][i,3] <- brier_est_fun(sim3_comb_women)
  brier_est_mc[[3]][i,4] <- nrow(sim3_women[sim3_women$AGE>=30 & sim3_women$AGE<=74,])
```

```
}

# Save and load simulation results
#saveRDS(brier_est_mc[[1]], "brier_est1.RDS")
#saveRDS(brier_est_mc[[2]], "brier_est2.RDS")
#saveRDS(brier_est_mc[[3]], "brier_est3.RDS")

brier_est_mc1 <- readRDS("brier_est1.RDS")
brier_est_mc2 <- readRDS("brier_est2.RDS")
brier_est_mc3 <- readRDS("brier_est3.RDS")

p_rm_unif <- ggplot(data=brier_est_mc1) +
  geom_line(aes(x=1:1000, y=cumsum(Men)/seq_along(1:1000)), col="forestgreen") +
  geom_line(aes(x=1:1000, y=cumsum(Women)/seq_along(1:1000)), col="orange") +
  labs(title="Uniform(1, 75)",
       x = "Number of Simulation", y="Estimated Brier Score") +
  ylim(0, 0.15) + theme_minimal()

p_rm_beta <- ggplot(data=brier_est_mc2) +
  geom_line(aes(x=1:1000, y=cumsum(Men)/seq_along(1:1000)), col="forestgreen") +
  geom_line(aes(x=1:1000, y=cumsum(Women)/seq_along(1:1000)), col="orange") +
  labs(title="Beta(10, 19)",
       x = "Number of Simulation", y="Estimated Brier Score") +
  ylim(0, 0.15) + theme_minimal()

colors <- c("Men" = "forestgreen", "Women" = "orange")
p_rm_norm <- ggplot(data=brier_est_mc3) +
  geom_line(aes(x=1:1000, y=cumsum(Men)/seq_along(1:1000), color="Men")) +
  geom_line(aes(x=1:1000, y=cumsum(Women)/seq_along(1:1000), color="Women")) +
  labs(title="Truncated Normal Distribution",
       x = "Number of Simulation", y="Estimated Brier Score", color="Gender") +
  scale_color_manual(values=colors) +
  ylim(0, 0.15) + theme_minimal()

plot_grid(p_rm_unif, p_rm_beta, p_rm_norm,
          rel_widths = c(1,1,1.4), ncol=3)

sim_sum1 <- data.frame(name = c("Mean", "SE", "Mean", "SE"),
           sim1 = c(mean(brier_est_mc1[,1]), sd(brier_est_mc1[,1]),
                    mean(brier_est_mc1[,3]), sd(brier_est_mc1[,3])),
           sim2 = c(mean(brier_est_mc2[,1]), sd(brier_est_mc2[,1]),
                    mean(brier_est_mc2[,3]), sd(brier_est_mc2[,3])),
           sim3 = c(mean(brier_est_mc3[,1]), sd(brier_est_mc3[,1]),
                    mean(brier_est_mc3[,3]), sd(brier_est_mc3[,3])))
rownames(sim_sum1) <- c("Men", "", "Women", " ")
sim_sum1[,2:4] <- round(sim_sum1[,2:4], 4)

sim_sum1 %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Average Estimated Brier Scores of the Model
      in Simulated Population, Three Independent Simulations",
      col.names = linebreak(c("", "Simulated NHANES (Uniform)",
                              "Simulated NHANES (Beta)",
```

```
                              "Simulated NHANES (Truncated Normal)")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'))%>%
  column_spec(1:5, width = "6em")

# Covariance
framingham_sim <- framingham_df %>%
  mutate(log_TOTCHOL=log(TOTCHOL),
         log_SYSBP=log(SYSBP),
         log_HDLC=log(HDLC),
         log_AGE=log(AGE)) %>%
  dplyr::select(SEX, log_TOTCHOL, log_SYSBP, log_HDLC, log_AGE,
                CURSMOKE, DIABETES, BPMEDS)

cov_men <- framingham_sim %>%
  filter(SEX==1) %>%
  dplyr::select(-c(SEX)) %>%
  cov()

cov_women <- framingham_sim %>%
  filter(SEX==2) %>%
  dplyr::select(-c(SEX)) %>%
  cov()

# Mean
nhanes_mean <- df_2017 %>%
  filter(AGE>=30 & AGE<=74) %>%
  group_by(SEX) %>%
  summarise(
    # 4 continuous -> log-scale
    TOTCHOL = mean(log(TOTCHOL), na.rm = TRUE),
    SYSBP = mean(log(SYSBP), na.rm = TRUE),
    HDLC = mean(log(HDLC), na.rm = TRUE),
    AGE = mean(log(AGE), na.rm = TRUE),
    # 3 categorical (binary) -> treat as continuous for simualtion
    CURSMOKE = mean(CURSMOKE, na.rm = TRUE),
    DIABETES = mean(DIABETES, na.rm = TRUE),
    BPMEDS = mean(BPMEDS, na.rm = TRUE)
  )

# Mean values for 7 variables, respectively
mean_men <- unlist(as.vector(nhanes_mean[1, -1]))
mean_women <- unlist(as.vector(nhanes_mean[2, -1]))

# Function: simulation using mvrnorm()
sim_fun_mv <- function() {
  n_sim_men <- 1961
  n_sim_women <- 2099

  # Multivariate normal distribution
  sim_men <- data.frame(mvrnorm(n_sim_men, mean_men, cov_men))
  sim_women <- data.frame(mvrnorm(n_sim_women, mean_women, cov_women))
```

```r
  # Convert binary variables back using quantiles
  sim_men <- sim_men %>%
    mutate(CURSMOKE = ifelse(CURSMOKE>quantile(CURSMOKE, unname(mean_men[5])), 0, 1),
           DIABETES = ifelse(DIABETES>quantile(DIABETES, unname(mean_men[6])), 0, 1),
           BPMEDS = ifelse(BPMEDS>quantile(BPMEDS, unname(mean_men[7])), 0, 1),
           SEX = 1)
  sim_men$TOTCHOL <- exp(sim_men$TOTCHOL)
  sim_men$SYSBP <- exp(sim_men$SYSBP)
  sim_men$HDLC <- exp(sim_men$HDLC)
  sim_men$AGE <- exp(sim_men$AGE)

  sim_women <- sim_women %>%
    mutate(CURSMOKE = ifelse(CURSMOKE>quantile(CURSMOKE, unname(mean_women[5])), 0, 1),
           DIABETES = ifelse(DIABETES>quantile(DIABETES, unname(mean_women[6])), 0, 1),
           BPMEDS = ifelse(BPMEDS>quantile(BPMEDS, unname(mean_women[7])), 0, 1),
           SEX = 2)
  sim_women$TOTCHOL <- exp(sim_women$TOTCHOL)
  sim_women$SYSBP <- exp(sim_women$SYSBP)
  sim_women$HDLC <- exp(sim_women$HDLC)
  sim_women$AGE <- exp(sim_women$AGE)

  return(list(sim_men, sim_women))
}

brier_est_mc4 <- data.frame(matrix(ncol=2, nrow=0,
                                   dimnames=list(NULL, c("Men", "Women"))))

for (i in 1:1000){
  set.seed(i+1)
  # Men
  sim_men <- sim_fun_mv()[[1]]
  sim_comb_men <- comb_fun(sim_men, "men")
  brier_est_mc4[i,1] <- brier_est_fun(sim_comb_men)

  # Women
  sim_women <- sim_fun_mv()[[2]]
  sim_comb_women <- comb_fun(sim_women, "women")
  brier_est_mc4[i,2] <- brier_est_fun(sim_comb_women)
}

# Save and load simulation results
#saveRDS(brier_est_mc4, "brier_est4.RDS")

brier_est_mc4 <- readRDS("brier_est4.RDS")

colors <- c("Men" = "forestgreen", "Women" = "orange")
ggplot(data=brier_est_mc4) +
  geom_line(aes(x=1:1000, y=cumsum(Men)/seq_along(1:1000), color="Men")) +
  geom_line(aes(x=1:1000, y=cumsum(Women)/seq_along(1:1000), color="Women")) +
  labs(title="Multivariate Normal Distribution",
       x = "Number of Simulation", y="Estimated Brier Score", color="Gender") +
  scale_color_manual(values=colors) +
  ylim(0, 0.15) + theme_minimal()
```

```r
sim_sum2 <- data.frame(name = c("Mean", "SE", "Mean", "SE"),
           sim1 = c(mean(brier_est_mc4[,1]), sd(brier_est_mc4[,1]),
                    mean(brier_est_mc4[,2]), sd(brier_est_mc4[,2])))
rownames(sim_sum2) <- c("Men", "", "Women", " ")
sim_sum2[,2] <- round(sim_sum2[,2], 4)

sim_sum2 %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Average Estimated Brier Scores of the Model
      in Simulated Population, Multivariate Normal",
      col.names = linebreak(c("", "Simulated NHANES (Multivariate Normal)")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'))

est_avg <- data.frame(framingham = c(mean((pred_men-test_men$CVD)^2),
                                     mean((pred_women-test_women$CVD)^2)),
           nhanes_drop = c(brier_est1[1,1], brier_est1[1,2]),
           nhanes_mi = c(mean(brier_est2[,1]), mean(brier_est2[,2])),
           nhanes_sim1 = c(mean(brier_est_mc1[,1]), mean(brier_est_mc1[,3])),
           nhanes_sim2 = c(mean(brier_est_mc2[,1]), mean(brier_est_mc2[,3])),
           nhanes_sim3 = c(mean(brier_est_mc3[,1]), mean(brier_est_mc3[,3])),
           nhanes_sim4 = c(mean(brier_est_mc4[,1]), mean(brier_est_mc4[,2])))
rownames(est_avg) <- c("Men", "Women")

round(est_avg, 4) %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Brier Scores and Average Estimated Brier Scores of the Model
      in Different Population",
      col.names = linebreak(c("Framingham",
                              "NHANES (Drop NA)",
                              "NHANES (MI)",
                              "Simulated NHANES (Uniform)",
                              "Simulated NHANES (Beta)",
                              "Simulated NHANES (Truncated Normal)",
                              "Simulated NHANES (Multivariate Normal)")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'),
                font = 7) %>%
  column_spec(1:8, width = "6em")
```