PHP2550 Project 3: Simulation Studies

Evaluating the Performance of a Prediction Model in Different Population

Yiwen Liang

12/03/2023

Abstract

This project investigates the transportability of a cardiovascular disease (CVD) prediction model developed on Framingham Heart Study data to a distinct target population represented by the 2017 National Health and Nutrition Examination Survey (NHANES). Three comprehensive evaluations are conducted to gauge the model's performance. Firstly, the Brier scores are computed within the original population of the Framingham study. Secondly, an alternative Brier score estimator proposed by Dr. Jon Steingrimsson is employed to assess the model's performance in the NHANES population. Lastly, a simulated target population is created based on summary statistics, and the Brier score estimator 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 indicate that the prediction model exhibits reasonable accuracy for CVD risk prediction, with better performance in females. Moreover, the model demonstrates robust transportability to the NHANES population. Limitations include exclusive reliance on Brier scores and potential refinements in the simulation methodology. This study contributes valuable insights into the model's generalizability, emphasizing its practical implications for healthcare professionals in mitigating cardiovascular risks.

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 evaluate the performance of a prediction model developed on the population in the Framingham Heart Study, in a different population, the data in 2017 from National Health and Nutrition Examination Survey (NAHNES), thereby offering valuable insights into the generalizability of this model. The project includes three 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. The last evaluation is similar to the second one, but in this case, we simulate the target population using only summary statistics when individual-level data is not available. By comparing the Brier scores and estimates from these three evaluations, the project aims to determine how well this prediction model perform in the target population and assess its overall transportability.

The data sources used in this project include data from `Framingham` and `NHANES` in 2017, and both are publicly available (in package `riskCommunicator` and `nhanesA`). 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_Project_3.

Methods

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.

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).

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.

Models

The model for the evaluation can be expressed as follows:

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

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).

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.

We'll 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.

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

where S is the population indicator ($S = 1$ if from `Framingham`, $S = 0$ if from NHANES), D is the test-train indicator ($D = 1$ if in training set, $D = 0$ 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:

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

Monte Carlo Simulation

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

- **Aims:** Evaluating the performance of the given model in a different population from the one in which it was originally developed.
- **Data-generating mechanisms:** Data are simulated on $n_{men} = 4557$ and $n_{women} = 4697$ (number of male and female participants in `NHANES`). Initially, we 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). 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);
3. Truncated Normal Distribution, with lower bound = 1, and upper bound = 75.

The distributions and specified parameters for the other six are presented in Table 1. How we determine them and the dimension of the dataset will be illustrated in the **Results** section, along with the summary of the two datasets.

Table 1: Distributions for Simulation

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)

- **Estimands:** Estimated Brier score using Equation (1).
- **Methods:** For each simulated dataset, we combine it with the `Framingham` dataset and then compute the estimated Brier score.
- **Performance measures:** Including convergence, number of observations after subsetting according to the criterion, and empirical standard errors.

Results

Performance Evaluation in Framingham

We begin by evaluating the model's performance within the `Framingham` dataset. The summary of the `Framingham` data stratified by gender is provided in Table 2. 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 2: Summary of the Variables in Framingham Dataset

	Men (SEX=1)	Women (SEX=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	

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

Men	Women
0.199292	0.111829

Performance Evaluation in the Target Population (NHANES)

We proceed to evaluate the model's performance in the target population underlying NHANES. The summary of the `NHANES` data stratified by gender is provided in Table 4. 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.

Table 4: Summary of the Variables in NHANES Dataset

	Men (SEX=1)	Women (SEX=2)	p	test
n	4557	4697		
SEQN (mean (SD))	98363.83 (2677.38)	98296.19 (2665.73)	0.223	
SYSBP (mean (SD))	122.49 (18.71)	120.20 (21.09)	<0.001	
SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001	
AGE (mean (SD))	34.12 (25.75)	34.55 (25.25)	0.419	
BMI (mean (SD))	26.16 (7.63)	26.98 (8.80)	<0.001	
HDLC (mean (SD))	49.57 (13.53)	57.01 (14.94)	<0.001	
CURSMOKE (mean (SD))	0.21 (0.41)	0.14 (0.35)	<0.001	
BPMEDS (mean (SD))	0.28 (0.45)	0.28 (0.45)	0.917	
TOTCHOL (mean (SD))	176.68 (40.38)	182.94 (40.59)	<0.001	
DIABETES (mean (SD))	0.11 (0.31)	0.09 (0.29)	0.001	

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

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. `SYSBP` exhibits the highest proportion of missing data, with 14.89% for men and 16.82% 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 5: Summary of Missing Values in NHANES

Variables	Men		Women	
	N	Proportion (%)	N	Proportion (%)
BMI	125	6.37	113	5.38
BPMEDS	**132**	**6.73**	**116**	**5.53**
DIABETES	**0**	**0.00**	**1**	**0.05**
HDLC	**218**	**11.12**	**209**	**9.96**
SYSBP	**292**	**14.89**	**353**	**16.82**
TOTCHOL	**218**	**11.12**	**209**	**9.96**

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

Men	Women
0.237703	0.167713

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

Men	Women
0.196904	0.127708
0.216186	0.138657
0.209197	0.129480
0.218307	0.118817
0.208430	0.129300
0.209805	**0.128792**

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. Table 4 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.

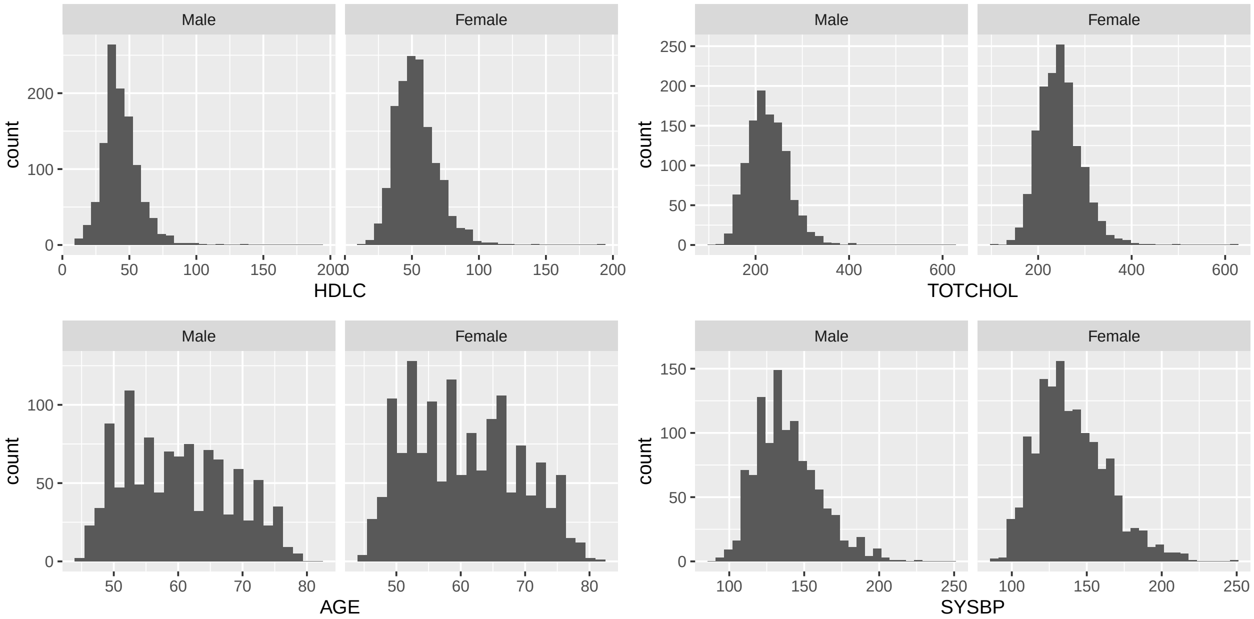

Figure 1: Distribution of Four Continuous Variables in Framingham Data

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 distributions of simulated HDLC, TOTCHOL, and SYSBP, along with the distribution of AGE using three approaches, are displayed in Figure 2 and Figure 3. Table 8 provides the summary statistics for simulated AGE with three distributions. Figure 2 demonstrates that our simulations for the three continuous variables are reasonable. However, based on Figure 3 and Table 8, there's no significant difference in the performance among the three distributions, allowing us to select any one of them. 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.

Table 8: Summary Statistics of AGE Simulation Using 3 Distributions

	NHANES	Uniform(1,75)	Beta(10,19)	Truncated Normal
Men	34.12 (25.75)	38.01 (21.4)	34.34 (8.71)	35.74 (18.58)
Women	34.55 (25.25)	38.18 (21.42)	34.44 (8.66)	36.19 (18.42)

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.

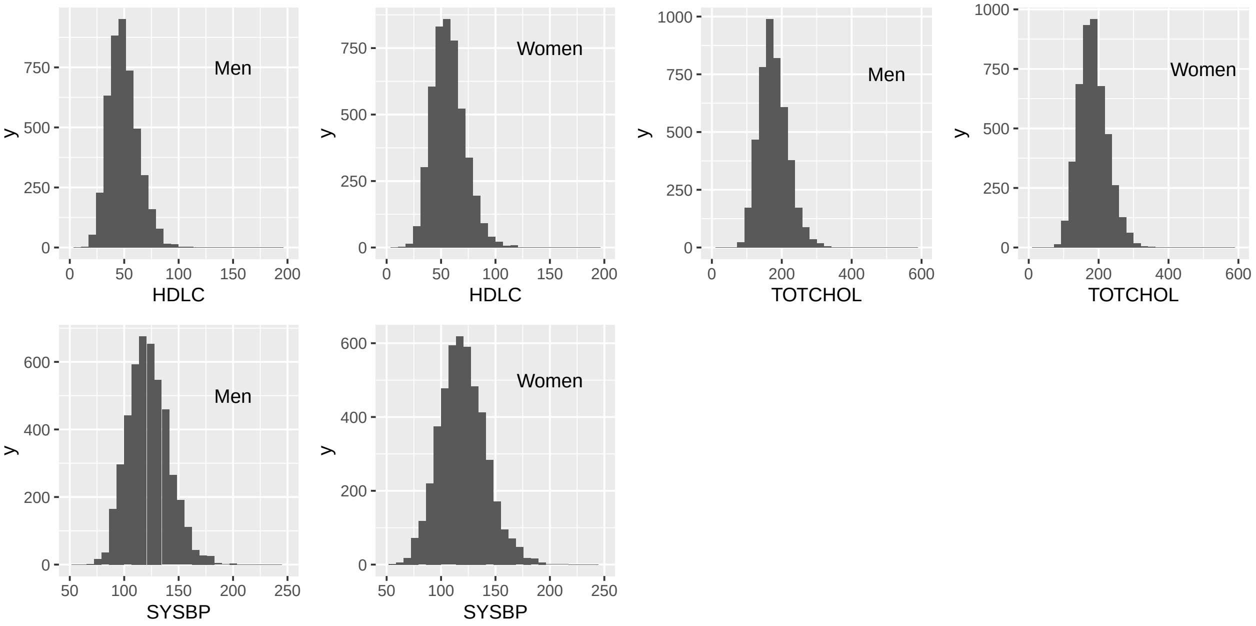

Figure 2: Distribution of Three Simulated Continuous Variables

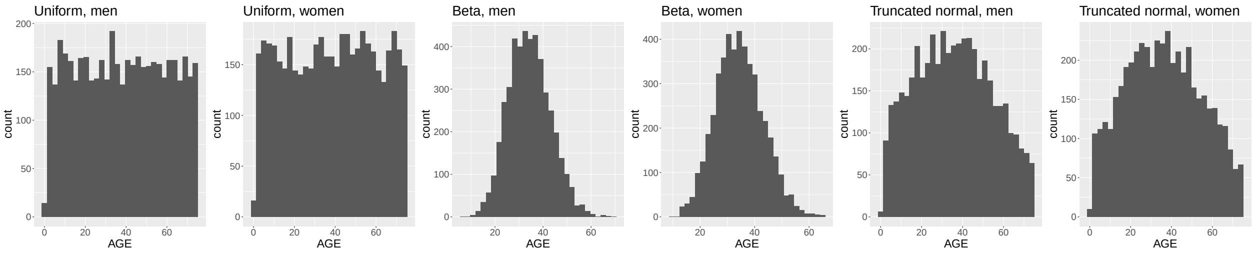

Figure 3: Distribution of Simulated Age Using Three Distributions

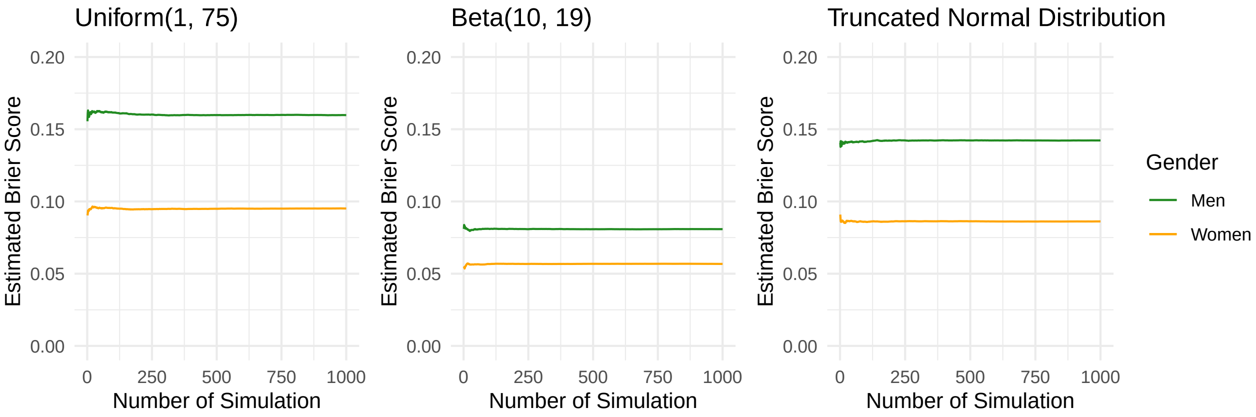

Figure 4: Cumualtive Average of the Estimates of Brier Scores for Three Simulations

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)
Men	0.199292	0.237703	0.209805	0.159797	0.080848	0.142238
Women	0.111829	0.167713	0.128792	0.095145	0.056779	0.086200

Figure 4 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 9, we present Brier scores measured within `Framingham` data, estimated Brier scores in the target population `NHANES` when we drop all missing entries and conduct multiple imputation (averaging estimates), and averaging estimated Brier scores in the simulated target population `NHANES` with 3 different distributions for `AGE`.

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 8 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.

Furthermore, according to Table 8, the Brier scores for the given model, when splitting the `Framingham` data into training and testing sets, are 0.199292 for males and 0.111829 for females. Considering these as the baseline for comparison, the estimated Brier scores in the target population `NHANES` are greater than the Brier risks obtained within the `Framingham` data, against which the models are constructed. This aligns with our expectations. Given the differences in demographic characteristics between the two population, it's reasonable and acceptable that the predictive accuracy of the model decreases when transported to a new population.

On the contrary, we observe that the estimated Brier scores in the simulated target population 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.

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. The observed increase in Brier score estimates falls within our acceptable expectations. 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.

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. Another limitation pertains to the simulation, especially `AGE`. Though the Brier score estimates from the simulated population seem reasonable and align with the research objectives, there's room for improvement in our simulation methodology. Enhancements could involve intentionally incorporating outliers, or simulating variables using alternative approaches, as opposed to relying solely on theoretical distributions.

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. The model is essential for estimating the risk of cardiovascular disease, which helps clinicians take precautions or reduce risks by modifying these risk factors at an early stage. Significant practical consequences result from the estimator of Brier scores, which enables researchers to evaluate the model's performance when applied to a different target population in the event that outcome 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.

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(knitr)
library(tidyverse)
library(riskCommunicator)
library(tableone)
library(nhanesA)
library(kableExtra)
library(ggplot2)
library(ggpubr)
library(reshape2)
library(cowplot)
library(mice)
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",
      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)

# Tableone for Framingham
tb1 <- print(CreateTableOne(data=framingham_df, strata = c("SEX")), printToggle = FALSE)
kable(tb1,
      caption = "Summary of the Variables in Framingham Dataset",
      col.names = linebreak(c("Men (SEX=1)", "Women (SEX=2)", "p", "test")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position')) %>%
  row_spec(c(4, 5, 6, 7, 9, 10, 11, 12) ,bold = TRUE)

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

# Looking at risk within 15 years - remove censored data
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

# 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$CVD)^2), 6),
           Women = round(mean((pred_women-test_women$CVD)^2), 6)) %>%
  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'))

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

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

# Predictions
framingham_df_men$pred <- predict(mod_men, type = "response")
framingham_df_women$pred <- predict(mod_women, type = "response")

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

# 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)
```

```r
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")

# Tableone for NHANES
tb2 <- print(CreateTableOne(data = df_2017, strata = c("SEX")), printToggle = FALSE)
kable(tb2,
      caption = "Summary of the Variables in NHANES Dataset",
      col.names = linebreak(c("Men (SEX=1)", "Women (SEX=2)", "p", "test")),
      row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
  kable_styling(full_width = FALSE,
                latex_options = c('striped', 'HOLD_position'))

df_2017 <- df_2017[df_2017$AGE>=30 & df_2017$AGE<=74,]

# 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()

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')) %>%
  row_spec(2:6, bold = TRUE)

# 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(df, sex="men") {
  df$SYSBP_UT <- ifelse(df$BPMEDS==0, df$SYSBP, 0)
  df$SYSBP_T <- ifelse(df$BPMEDS==1, df$SYSBP, 0)

  df <- df %>%
    dplyr::select(c("SEX", "HDLC", "TOTCHOL", "AGE", "SYSBP_UT",
                  "SYSBP_T", "CURSMOKE", "DIABETES")) %>%
    mutate(S = 0, CVD = NA)

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

  df_comb$D <- sample(c(1, 0), nrow(df_comb),
                      replace=TRUE, prob=c(0.7,0.3))

  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)
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(df) {
  m_o <- glm(S ~ D + log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT+1)
             + log(SYSBP_T+1) + CURSMOKE + DIABETES,
             family = "binomial", data = df)

  df$o_hat <- 1/predict(m_o, type="response")

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

  return(sum(df_temp$o_hat*(df_temp$CVD-df_temp$pred)^2) / sum(df$S==0 & df$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'))

# 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)

label_sex <- c(`1` = "Male", `2` = "Female")

# HDLC
hist_HDLC <- ggplot(data=df_eda) +
  geom_histogram(aes(HDLC)) +
  facet_wrap(~ SEX, labeller = as_labeller(label_sex))

# TOTCHOL
hist_TOTCHOL <- ggplot(data=df_eda) +
  geom_histogram(aes(TOTCHOL)) +
  facet_wrap(~ SEX, labeller = as_labeller(label_sex))

# AGE
hist_AGE <- ggplot(data=df_eda) +
  geom_histogram(aes(AGE), bins=25) +
  facet_wrap(~ SEX, labeller = as_labeller(label_sex))

# SYSBP
hist_SYSBP <- ggplot(data=df_eda) +
  geom_histogram(aes(SYSBP)) +
  facet_wrap(~ SEX, labeller = as_labeller(label_sex))

plot_grid(hist_HDLC, hist_TOTCHOL, hist_AGE, hist_SYSBP, ncol=2)

# 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)
```

```r
# 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
```

```r
  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))
}

# Visualize simulated distributions
set.seed(2550)
sim_men <- sim_fun("uniform")[[1]]
sim_women <- sim_fun("uniform")[[2]]

# HDLC
hist_HDLC_men <- ggplot(data=sim_men) +
  geom_histogram(aes(HDLC)) + xlim(0,200) +
  annotate("text", x=150, y=750, label="Men")
hist_HDLC_women <- ggplot(data=sim_women) +
  geom_histogram(aes(HDLC)) + xlim(0,200) +
  annotate("text", x=150, y=750, label="Women")

# TOTCHOL
hist_TOTCHOL_men <- ggplot(data=sim_men) +
  geom_histogram(aes(TOTCHOL)) + xlim(0,600) +
  annotate("text", x=500, y=750, label="Men")
hist_TOTCHOL_women <- ggplot(data=sim_women) +
  geom_histogram(aes(TOTCHOL)) + xlim(0,600) +
  annotate("text", x=500, y=750, label="Women")

# SYSBP
hist_SYSBP_men <- ggplot(data=sim_men) +
  geom_histogram(aes(SYSBP)) + xlim(50,250) +
  annotate("text", x=200, y=500, label="Men")
hist_SYSBP_women <- ggplot(data=sim_women) +
  geom_histogram(aes(SYSBP)) + xlim(50,250) +
  annotate("text", x=200, y=500, label="Women")

plot_grid(hist_HDLC_men, hist_HDLC_women,
          hist_TOTCHOL_men, hist_TOTCHOL_women,
          hist_SYSBP_men, hist_SYSBP_women, ncol=4)

# Visualize simulated distributions
sim_age <- data.frame(matrix(ncol=4, nrow=2,
                             dimnames=list(NULL, c("NHANES", "s1", "s2", "s3"))))
rownames(sim_age) <- c("Men", "Women")
sim_age[,1] <- c("34.12 (25.75)", "34.55 (25.25)")

# AGE1
set.seed(2550)
```

```r
sim_men <- sim_fun("uniform")[[1]]
sim_women <- sim_fun("uniform")[[2]]
sim_age[,2] <- c(paste(round(mean(sim_men$AGE),2), ' (', round(sd(sim_men$AGE),2), ')'),
                 paste(round(mean(sim_women$AGE),2), ' (', round(sd(sim_women$AGE),2), ')'))
hist_AGE1_men <- ggplot(data=sim_men) +
  geom_histogram(aes(AGE)) +
  ggtitle("Uniform, men") +
  theme(text = element_text(size=20))
hist_AGE1_women <- ggplot(data=sim_women) +
  geom_histogram(aes(AGE)) +
  ggtitle("Uniform, women") +
  theme(text = element_text(size=20))

# AGE2
set.seed(2550)
sim_men <- sim_fun("beta")[[1]]
sim_women <- sim_fun("beta")[[2]]
sim_age[,3] <- c(paste(round(mean(sim_men$AGE),2), ' (', round(sd(sim_men$AGE),2), ')'),
                 paste(round(mean(sim_women$AGE),2), ' (', round(sd(sim_women$AGE),2), ')'))
hist_AGE2_men <- ggplot(data=sim_men) +
  geom_histogram(aes(AGE)) +
  ggtitle("Beta, men") +
  theme(text = element_text(size=20))
hist_AGE2_women <- ggplot(data=sim_women) +
  geom_histogram(aes(AGE)) +
  ggtitle("Beta, women") +
  theme(text = element_text(size=20))

# AGE3
set.seed(2550)
sim_men <- sim_fun("normal")[[1]]
sim_women <- sim_fun("normal")[[2]]
sim_age[,4] <- c(paste(round(mean(sim_men$AGE),2), ' (', round(sd(sim_men$AGE),2), ')'),
                 paste(round(mean(sim_women$AGE),2), ' (', round(sd(sim_women$AGE),2), ')'))
hist_AGE3_men <- ggplot(data=sim_men) +
  geom_histogram(aes(AGE)) +
  ggtitle("Truncated normal, men") +
  theme(text = element_text(size=20))
hist_AGE3_women <- ggplot(data=sim_women) +
  geom_histogram(aes(AGE)) +
  ggtitle("Truncated normal, women") +
  theme(text = element_text(size=20))

plot_grid(hist_AGE1_men, hist_AGE1_women,
          hist_AGE2_men, hist_AGE2_women,
          hist_AGE3_men, hist_AGE3_women, ncol=6)

sim_age %>%
  mutate_all(linebreak) %>%
  kbl(caption = "Summary Statistics of AGE Simulation Using 3 Distributions",
      col.names = linebreak(c("NHANES", "Uniform(1,75)",
                              "Beta(10,19)", "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, bold = TRUE)

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, 0.2) + 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, 0.2) + 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, 0.2) + theme_minimal()

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

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])))
rownames(est_avg) <- c("Men", "Women")

round(est_avg, 6) %>%
  mutate_all(linebreak) %>%
```

```r
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)")),
    row.names = TRUE, booktabs = TRUE, escape = TRUE, align = "c") %>%
kable_styling(full_width = FALSE,
              latex_options = c('striped', 'HOLD_position')) %>%
column_spec(1:7, width = "6em")
```