Evaluating Prediction Model Performance across Different Simulated Populations

Monica Colon-Vargas

Fall 2023

Objective: The study primarily aims to assess the performance of the cardiovascular risk prediction model fitted on the Framingham dataset in a specific demographic derived from the NHANES dataset. The study aims to compare the model transportability in simulated populations, enabling a comparison of the model's performance across different demographic scenarios.
Methods: To evaluate performance we use the estimation of the Brier score proposed in [4] when the target population lacks the outcome variable. The simulation is given in the ADEMP framework from [3]. The Brier's estimate is calculated in different simulated populations varying the correlation of the variables.
Conclusions: In analyzing the NHANES dataset, the estimated Brier score differed by gender, being lower in men compared to the source population, possibly due to a healthier male cohort. Simulated data for NHANES showed higher Brier scores for both genders, suggesting poor model transportability, especially for women, although estimates were precise. The standard errors were low, indicating stable and consistent estimations. However, the moderate Brier score of around 0.3 suggests some inaccuracy in the model's predictions, warranting potential model adjustments for improved accuracy in the target population.

1 Introduction

The development of predictive models plays a crucial role in healthcare decision-making and risk assessment. However, ensuring the efficacy of these models across diverse populations—known as model transportability—remains a critical challenge. The ability of a predictive model developed in one population to reliably and effectively perform when applied to a different yet related population is fundamental for its real-world applications. However, the data used to develop these prediction models, often derived from randomized trials, observational databases, or prospective cohort studies, might not accurately represent the characteristics of the intended target population. For instance, healthcare systems might seek to implement a risk prediction model to identify high-risk individuals for cardiovascular events among their patient cohort. However, the data used to develop these prediction models, often derived from randomized trials, observational databases, or prospective cohort studies, might not accurately represent the characteristics of the intended target population. This demands of sophisticated methodologies, rigorous validation techniques, and calibration strategies to bridge the gap between the source population used for model development and the intended target population. Strategies to assess and enhance model transportability can be employed. This includes techniques like external validation, where the model is tested on an independent dataset representing the target population, and calibration methods that adjust model predictions to match the characteristics of the new population. However, the new target population often lacks the pertinent outcome under investigation.

Issues with transportability arise due to discrepancies between the source population used for model development and the target population where the model will be deployed. Factors like differences in demographics, disease prevalence, lifestyle, genetic makeup, or healthcare practices can influence how well a model performs in a new setting. In this scenario, the absence of outcome details in the target population prevents creating or evaluating prediction models solely using data from that specific population. Consequently, relying on both covariate and outcome data from the source population might present an appealing option, given that adjustments can be made to account for variations in data distributions between the two populations [4].

This investigation assesses Framingham Dataset models' ability to predict cardiovascular disease in NHANES, which lacks specific outcome data. We will apply separate gender-based models, and explore their performance in a different dataset without complete outcome information. This approach aims to understand if these models remain effective across diverse populations, offering insights into their broader applicability.

The study's primary objective is to evaluate the efficacy of a cardiovascular prediction model within a specified demographic drawn from the NHANES dataset. The emphasis lies in evaluating the model's predictive capacity for cardiovascular risk within this particular population. The study calculates model performance for simulated populations underlying the NHANES population, allowing for comparative analysis of the model's performance under varying demographic scenarios.

2 Data

In this investigation, the utilization of the Framingham Dataset to predict cardiovascular disease (CVD) within a distinct population, namely the NHANES dataset, is proposed. However, the NHANES dataset lacks the relevant outcome information for CVD. The primary objective is to gauge the model's efficacy when applied to a dissimilar dataset. Specifically, two distinct models will be developed using the Framingham Dataset, stratified by gender (male and female), in an endeavor to assess and compare their predictive performance in the absence of outcome data within the NHANES dataset. This approach aims to handle the challenge of using models designed for one group on a different population without the necessary outcome data. Our focus is on how well these models work within the NHANES dataset despite the missing outcome specifics. By doing this, we hope to understand if these models remain useful across different groups of people. This investigation could reveal whether these predictive tools keep their effectiveness when dealing with diverse populations, offering valuable insights into their broader usefulness and relevance.

Aligning the Framingham and NHANES datasets initiates with data preprocessing. The goal is to identify and extract variables from NHANES that mirror those utilized in the Framingham models. This involves a selection process to ensure similarity in variables used for modeling. Additionally, to facilitate direct comparisons, adjustments are made within the NHANES dataset, specifically focusing on constraining age parameters to match the age ranges observed in Framingham. These efforts aim to establish a consistent framework between the datasets, enabling a more accurate and meaningful comparative analysis of their respective predictive models.

2.1 Framingham Dataset

The Framingham Heart Study, initiated in 1948, is a pioneering long-term, multigenerational study designed to investigate the causes of heart disease. Conducted in Framingham, Massachusetts, it aimed to identify common factors that contribute to cardiovascular disease (CVD). The study enrolled over 5,000 participants initially, comprising individuals from diverse socio-economic backgrounds. [2] The Original Cohort, founded in 1948, consisted of 5,209 men and women. Requirements for entry were an age between 30 and 62 years at the time of first examination, with no history of heart attack or stroke.

The primary objectives included identifying risk factors for heart disease and strokes by observing participants over an extended period. This landmark study established the significance of various factors such as high blood pressure, high cholesterol levels, smoking, obesity, and other lifestyle and physiological traits as contributors to cardiovascular health. The study's findings significantly influenced public health recommendations and laid the groundwork for modern cardiovascular risk assessment and prevention strategies. The Framingham Heart Study reshaped healthcare priorities in the latter 20th century. They steered focus from treating existing cardiovascular disease to preventing it in at-risk individuals. Identifying those prone to future heart issues became crucial, enabling targeted preventive measures—a shift from reactive treatment to proactive prevention in cardiovascular health. Table 1 displays the summary statistics of the Framingham Dataset.

Table 1: Summary Statistics of Framingham Dataset

Variable	**N**	**Overall**,N=4,060	**Male**,N=2099	**Female**,N=1961
Continuous Variables				
HDLC	2,539	49.01 (15.45)	43.63 (13.37)	53.07 (15.67)
TOTCHOL	2,539	237.76 (44.91	226.44 (41.49)	246.32 (45.51)
AGE	2,539	60.32 (8.31)	60.01 (8.18)	60.55 (8.40)
SYSBP	2,539	139.51 (22.54)	138.94 (20.89)	139.94 (23.71)
Categorical Variables				
CURSMOKE	2,539			
No		1,669 (65.73%)	669 (61.15%	1,000 (69.20%)
Yes		870 (34.27%)	425 (38.85%	445 (30.80%)
DIABETES	2,539			
No		2,348 (92.48%)	998 (91.22%	1,350 (93.43%)
Yes		191 (7.52%)	96 (9.78%	95 (6.57%)
BPMEDS	2,539			
No		2,157 (84.95%)	971 (88.76%	1,186 (82.08%)
Yes		382 (15.05%)	123 (11.24%	259 (17.92%)

Note: Summary statistics for variables by sex. Continuous variables display mean and standard deviation (sd). Discrete variables display count and percentage (%)

2.2 Nhanes Dataset

The National Health and Nutrition Examination Survey (NHANES) serves as a vital source of comprehensive health and nutritional data in the United States. Conducted by the CDC, NHANES stands as a population-based survey crafted to gather comprehensive insights into the health and nutritional aspects of the United States household population released in two-year cycles. This survey operates through two distinct phases:

an in-depth home interview and a comprehensive health examination.

In the home interview segment, participants respond to inquiries encompassing health status, medical background, and dietary habits. Meanwhile, the health examination phase involves a series of meticulous medical and dental evaluations, precise physiological measurements, and extensive laboratory tests meticulously conducted by proficient and extensively trained medical personnel. This multifaceted approach ensures a comprehensive understanding of the participants' health profiles, fostering a robust dataset for in-depth health and nutritional analyses. This survey captures a broad spectrum of health-related information, spanning demographics, health indicators, dietary habits, physical assessments, and laboratory analyses. It offers insights into participants' demographics, health conditions like blood pressure and diabetes, detailed dietary intake, physical measurements such as height and weight, and an array of laboratory tests assessing various health markers [1]. Table 2 displays the summary statistics of the NHANES Dataset.

Table 2: Summary Statistics of NHANES Dataset

Variable	**N**	**Overall**,N=4,060	**Male**,N=2099	**Female**,N=1961
Continuous Variables				
HDLC	3,633	52.91 (15.79)	47.83 (14.07)	57.60 (15.84)
Unknown		427	218	209
TOTCHOL	3,633	192.17 (41.04)	188.40 (41.68)	195.64 (40.13)
Unknown		427	218	209
AGE	4,060	52.40 (12.64)	52.92 (12.71)	51.91 (12.56)
SYSBP	3,415	127.01 (19.07)	128.29 (17.67)	125.78 (20.24)
Unknown		645	292	353
Categorical Variables				
CURSMOKE	4,060			
No		3,249 (80.02%)	1,487 (75.83%	1,762 (83.94%)
Yes		811 (19.98%)	474 (24.17%	337 (16.06%)
DIABETES	4,059			
No		3,384 (83.37%)	1,603 (81.74%	1,781 (84.89%)
Yes		675 (16.63%)	358 (18.26%	317 (15.11%)
Unknown		1	0	1
BPMEDS	3,812			
No		2579 (67.65%)	1,225 (66.98%	1,354 (68.28%)
Yes		1,233 (32.35%)	604 (33.02%	629 (31.72%)
Unknown		248	132	116

Note: Summary statistics for variables by sex. Continuous variables display mean and standard deviation (sd). Discrete variables display count and percentage (%)

2.3 Missing Data

In both the Framingham and NHANES datasets, the presence of missing data poses a significant challenge. To assess the efficacy of models constructed using the Framingham dataset, a procedure of 5 multiple imputations is conducted on the NHANES dataset. Subsequently, the Brier score is estimated across the 5 imputed datasets, and the average of these results is computed.

3 Methodology

3.1 Model

Table 3 presents the variables incorporated into the model derived from the Framingham dataset. Initial modifications involve the creation of two novel variables delineating blood pressure status, contingent upon an individual's use of blood pressure medication (BPMEDS). Subsequently, logarithmic transformations are applied to all continuous variables. Additionally, logistic regression models are fitted separately for both sexes, harnessing these enhanced variables as part of the modeling process. Both logistic models are set to be applied across distinct simulated populations, to explore its transferability.

Variables	**Description**	**Type**
HDLC	High-Density Lipoprotein Cholesterol (mg/dL)	Continuous
TOTCHOL	Serum Total Cholesterol (mg/dL)	Continuous
AGE	Age at examination (years)	Continuous
SYSBP	Systolic Blood Pressure (mean of last two of three measurements) (mmHg)	Continuous
BPMEDS	Blood Pressure Medication Use	Discrete
CURSMOKE	Current Cigarette Smoking at Examination. 0 = Not current smoker, 1 = Current smoker	Discrete
DIABETES	Diabetic according to criteria of first exam treated or first exam with casual glucose of 200 mg/dL or more. 0 = Not diabetic, 1 = Diabetic	Discrete

Table 3: Variables used in both models stratified by sex

3.2 Metrics for Evaluation

We now shift our focus toward evaluating how well the model performs within the target population. Specifically, we concentrate on assessing the model's performance using the squared error loss function, aiming to determine and estimate its expectation, known as the mean squared error (MSE), within this population. This squared error loss, denoted as $(Y - g((X))^2$, where g is the fitted model, measures the difference between the observed outcome Y and the prediction derived from the model, g, represented as the square of their discrepancy. In [4], an estimator for the target population is given by:

$$\hat{\phi}_g = \frac{\sum_{i=1}^{n} I(S_i = 1, D_{test,i} = 1)\hat{o}(X_i)(Y_i - g(X_i))^2}{\sum_{i=1}^{n} I(S_i = 0), D_{test,i} = 1)} \tag{1}$$

Where the variable S is defined as 1 if the individual is from the Framingham dataset and 0 if the individual is part of the NHANES dataset. The $\hat{o}(X)$ is an estimator for the inverse-odds weights in the test set given by

$$\hat{o}(X) = \frac{P(S = 0|X, D_{test} = 1)}{P(S = 1|X, D_{test} = 1)}. \tag{2}$$

The numerator in equation (1) exclusively pertains to individuals within the Framingham dataset. The Brier score for the Framingham Dataset is calculated but involves weighing the contribution of each individual by their respective predicted outcome, denoted by $\hat{o}(X)$.

To use this estimator, two conditions need to be satisfied. The first condition connects the source and target populations, which can typically be a strong assumption. The first condition states the independence

of the outcome Y and the population S conditional on covariates. This requires the relationship between the outcome and covariates to be common across populations. The second condition states that every pattern of the covariates needed to satisfy the first condition can occur in the source data (the data that the model is fitted) [4] .

3.3 Simulation

We discuss how the simulation is developed using the ADEMP framework discussed in [3].

- **AIM:** The simulation aims to explore the impact of altering the correlation between variables while generating data based on fixed statistics derived from the NHANES dataset (such as mean and standard deviation). Specifically, this exploration aims to understand how these variations influence the estimation of the Brier score for the target population.

- **Data Generation Mechanism:** As depicted in Figure 1, the variables can be approximated utilizing a normal distribution based on the mean and variance of the log-transformed values. While certain approximations, like TOTCHOL, demonstrate high accuracy, others, such as AGE, exhibit less precise estimations. Nonetheless, our approach involves drawing multivariate random samples to simulate the logarithms of these variables. We will explore the estimation of the Brier score across four distinct correlation settings. These settings include independent variables, low correlation among variables, moderate correlation among variables, high correlation among variables, and a correlation pattern akin to that observed in the Framingham dataset. Figure 2 shows the different correlation structures used. In each iteration, samples of size $n = 3055$ (for men) and $n = 3544$ (for women) are drawn from a multinormal distribution utilizing the mean and standard deviation parameters obtained from transformed variables, and incorporating the mean and standard deviation of discrete variables. The choice of these sample sizes aligns with the respective number of male and female individuals within the NHANES dataset. This method, although producing continuous distributions for discrete variables, is rectified by employing a mean-quantile approach for each variable. The mean-quantile finds the sample quantile aligned with 1 minus the utilized simulation mean (different for each discrete variable). Values larger than this quantile will be set to 1 and 0 otherwise. This adaptation ensures discrete distributions, correlated with continuous ones, maintaining parity in mean values from the Framingham dataset.

- **Estimands:** The estimands under investigation in this simulation study are described in equation (1). The study aims to explore the impact of Brier score estimation on simulated populations, focusing on understanding how variations in the correlation affect this metric.

- **Methods:** This study encompasses two logistic regression models stratified by sex, aiming to predict cardiovascular disease (CVD). The variables employed in these models are detailed in Table 3. To compute the inverse-odds weights outlined in equation (2), a logistic regression model is utilized to predict $P(S = 1|X, D_{test} = 1)$.

- **Performance Measures:** To assess the impact of correlation on Brier Score estimation, our analysis involves the calculation of the bias between the estimated brier scores on the simulated data and the estimated brier score of the original NHANES dataset.

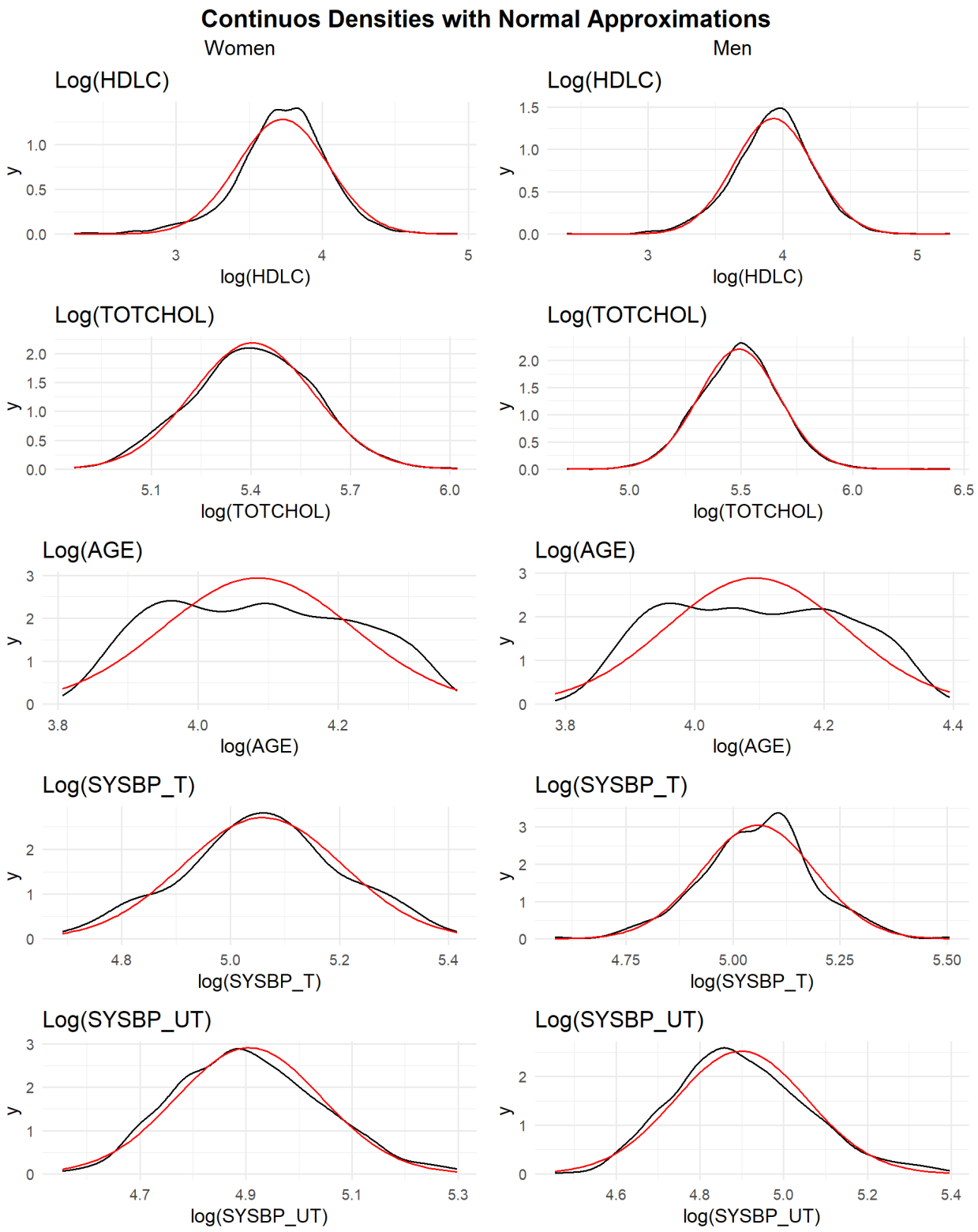

Figure 1: Approximations of log-transformed continuous variables using a normal distribution with mean and variance estimates derived from the logarithmically transformed variable

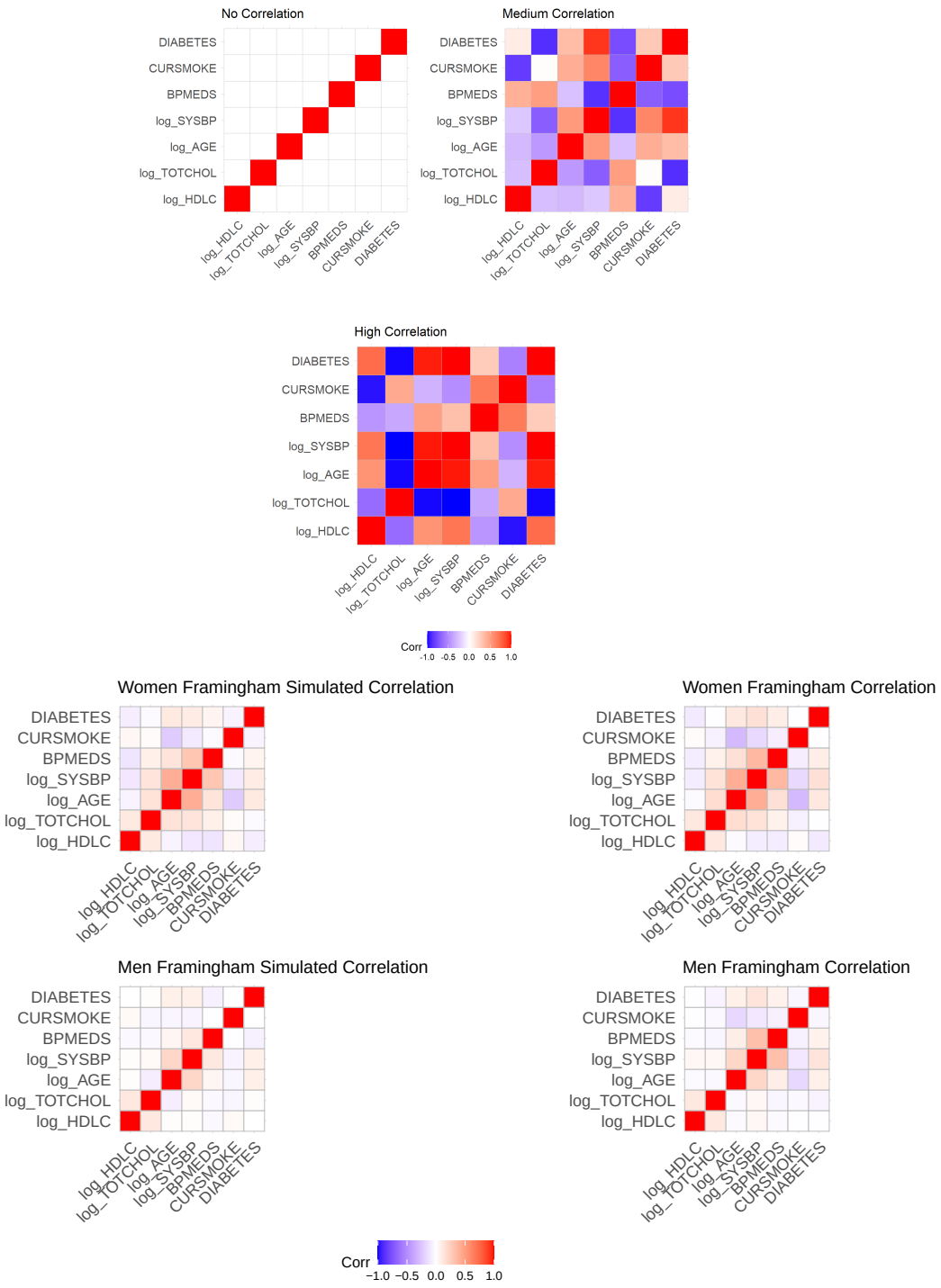

Figure 2: The initial three reflect uncorrelated, moderately correlated, and highly correlated variables. The subsequent four correlations are designed to mirror those observed in the Framingham dataset for men and women.

4 Results

The results for 5,000 simulations are presented in Figure 3 and in Table 4. Figure 3 illustrates the distributions concerning the estimated Brier scores across various simulation settings for both men and women. Notably, all distributions exhibit a semblance of normality, characterized by low standard deviations. In Table 4, the mean and standard deviation of the estimate of the bier score is calculated for each simulation setting for both men and women. The brier score estimate for men is lower than for women but the standard error for woman are lower than for men.

The model derived from the Framingham dataset yielded Brier scores of 0.1919 for men and 0.1160 for women. In essence, the Brier score of 0.1919 implies that the model's predicted probabilities for certain events might be less precise or less calibrated, leading to larger discrepancies between the predicted probabilities and the actual outcomes. Conversely, the Brier score of 0.1160 indicates that the model's predicted probabilities for a different set of events are more accurate or well-calibrated, resulting in smaller deviations between the predicted and observed outcomes. The estimated NHANES Brier scores were 0.0993 for men and 0.0506 for women. Both estimations exhibited a lower value compared to the score derived from the dataset where the model was fitted. This divergence might stem from the demographic disparities between the NHANES and Framingham datasets, suggesting a potential explanation: the NHANES dataset possibly represents a population with a healthier profile in contrast to the Framingham dataset.

The observed estimated Brier scores for simulated data are higher overall. For women, the Brier score remains low, indicating good predictive accuracy, while for men, it shows moderately good predictions. Among women, these scores converge around 0.11, peaking in scenarios mirroring a high correlation matrix, while lower values for moderately correlated variables. Conversely, for males, the estimated scores hover around 0.19. The highly correlated structure yielded higher estimated Brier scores, and a moderately correlated structure resulted in a lower estimated Brier score.

Table 5 shows the calculated bias for this estimated brier score where the true brier score is the estimated brier score from the NHANES dataset. We can conclude that we Brier score estimates are overestimating. However, the simulated datasets may not capture the real relationship occurring in the NHANES dataset which can explain this overestimation.

The observed trends highlight the influence of correlation structures on predictive accuracy, as indicated by the Brier scores. The Brier score estimates from equation (1) exhibit a close alignment between male and female scenarios, reflecting precision up to the third decimal place. The correlation patterns resembling those present in the Framingham dataset tend to yield slightly higher estimated Brier scores. This suggests that when the simulated data mimic the specific correlation structures seen in the Framingham dataset, the model's predictive performance, measured by the Brier score, tends to be less accurate or more uncertain. Additionally, when the correlation matrix exhibits high correlations, the estimated Brier scores are higher indicating that the model is worst calibrated in highly correlated scenarios.

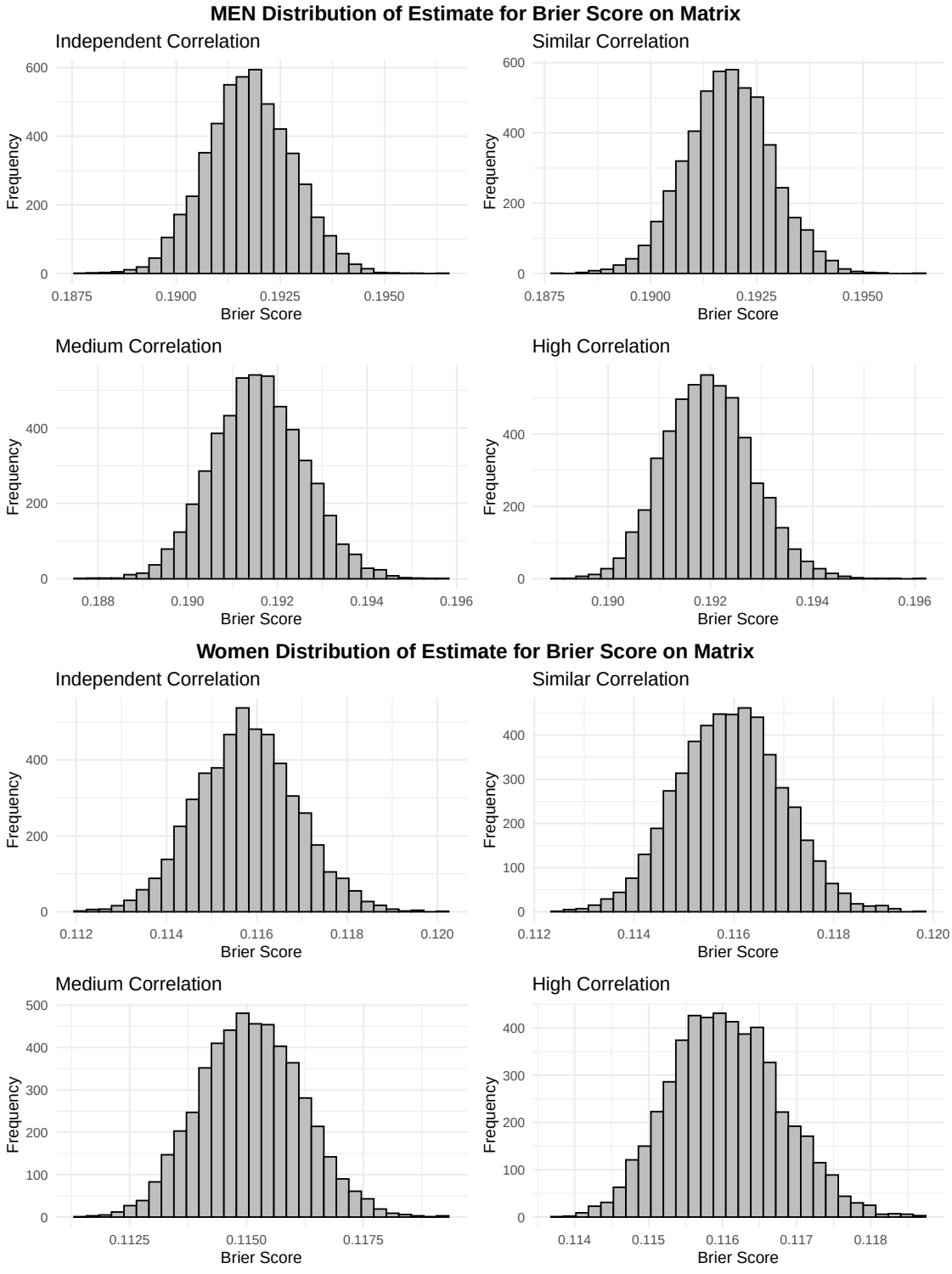

Figure 3: Distribution of the estimated Brier score

Model	Estimate for Brier	SD Brier
Uncorrelated	0.1158	0.0011
Similar	0.1159	0.0010
Medium	0.1151	0.0011
High	0.1161	0.0007

(a) Brier Results for Women Model
NHANES Estimate for Brier Score: 0.0506
Framingham Brier Score: 0.1160

Model	Estimate for Brier	SD Brier
Uncorrelated	0.1918	0.0010
Similar	0.1918	0.0010
Medium	0.1916	0.0010
High	0.1920	0.0090

(b) Brier Results for Men Model
NHANES Estimate for Brier Score: 0.0993
Framingham Brier Score: 0.1919

Table 4: Comparison of Brier Results for Women and Men Models. The Estimate for Brier is calculated as the mean of the estimated brier score calculated in each iteration

Model	Bias	
	Men	Women
Uncorrelated	0.0924	0.0.0652
Similar	0.0925	0.0654
Medium	0.0923	0.0645
High	0.0926	0.0655

Table 5: Bias Results for Men and Women Models. Bias is calculated as the difference between the estimated Brier score and the estimated NHANES Brier score.

5 Discussion

This study focused on estimating the brier score when applying a CVD prediction model stratified by sex on a different population from where the model was fitted. Particuarly, the outcome for CVD was not available in the target population making calculating the brier score nor possible. This is why we utilize the estimation of the Brier score when the target population lacks the outcome of the variable given in [4].

In analyzing the NHANES dataset, the estimated Brier was lower than in the source population. This could potentially be attributed to the relatively superior health status of male individuals within the NHANES population compared to the Framingham dataset. Furthermore, while evaluating the simulated data underlying NHANES, both men and women exhibited elevated Brier score estimates, with women demonstrating comparatively poorer performance. A plausable explanation for these elevated Brier scores can be that the simulated datasets were not capturing the true underlying relationship in the NHANES dataset. However, it's noteworthy that the estimates converge closely when varying the correlation matrix, indicating a level of precision in the estimation process. The standard errors of these estimations were observed to be low, implying a high degree of precision in the calculated estimates. This suggests that the estimation process exhibited a remarkable level of stability and consistency, offering reliable and robust results with minimal variation or uncertainty. Of particular significance is the close resemblance between the estimated Brier scores and the actual Brier scores derived from the Framingham dataset, observed across both male and female cohorts. This alignment strongly hints at the potential transportability of these models, indicating their promising adaptability and generalizability to novel contexts or diverse populations. Nonetheless, to enhance the model's performance within these simulated populations, tailoring the model specifically to these contexts could be made. The adaptation of tailoring the model to the target population might lead to a reduction in the estimated Brier score and a subsequent improvement in predictive accuracy.

A method proposed in [4] to tailor a possibly misspecified predictive model for use in the target population involves three sequential steps. Initially, an estimation of the likelihood of membership in the source population is conducted by merging the training data from both the source and target populations. Subsequently, these estimated probabilities are employed to create inverse-odds weights for each observation within the training set originating from the source population. Finally, these weights are utilized to derive the prediction model, employing all observations present in the training set from the source population.

References

[1] James A Fain. Nhanes: use of a free public data set, 2017.

[2] Syed S Mahmood, Daniel Levy, Ramachandran S Vasan, and Thomas J Wang. The framingham heart study and the epidemiology of cardiovascular disease: a historical perspective. *The lancet*, 383(9921):999–1008, 2014.

[3] Tim P Morris, Ian R White, and Michael J Crowther. Using simulation studies to evaluate statistical methods. *Statistics in medicine*, 38(11):2074–2102, 2019.

[4] Jon A Steingrimsson, Constantine Gatsonis, Bing Li, and Issa J Dahabreh. Transporting a prediction model for use in a new target population. *American Journal of Epidemiology*, 192(2):296–304, 2023.

Code Apendix

```r
library(riskCommunicator)
library(tidyverse)
library(tableone)
library(naniar)
library(dplyr)
library(MASS)
library(corrplot)
library(GGally)
library(ggpubr)
library(nhanesA)
library(ggcorrplot)
library(kableExtra)
```

```r
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)

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

#Factor variables
framingham_df <- framingham_df %>%
  mutate(SEX = as.factor(SEX),
         CURSMOKE = as.factor(CURSMOKE),
         DIABETES = as.factor(DIABETES),
         BPMEDS = as.factor(BPMEDS)
         )

# Filter to each sex
```

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

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

```r
    # The NHANES data here finds the same covariates among this national
       survey data


# blood pressure, demographic, bmi, smoking, and hypertension info
bpx_2017 <- nhanes("BPX_J") %>%
  dplyr::select(SEQN, BPXSY1 ) %>%
  rename(SYSBP = BPXSY1)
demo_2017 <- nhanes("DEMO_J") %>%
  dplyr::select(SEQN, RIAGENDR, RIDAGEYR) %>%
  rename(SEX = RIAGENDR, AGE = RIDAGEYR)
bmx_2017 <- nhanes("BMX_J") %>%
  dplyr::select(SEQN, BMXBMI) %>%
  rename(BMI = BMXBMI)
smq_2017 <- nhanes("SMQ_J") %>%
  mutate(CURSMOKE = case_when(SMQ040 %in% c(1,2) ~ 1,
                              SMQ040 == 3 ~ 0,
                              SMQ020 == 2 ~ 0)) %>%
  dplyr::select(SEQN, CURSMOKE)
bpq_2017 <- nhanes("BPQ_J") %>%
  mutate(BPMEDS = case_when(
    BPQ020 == 2 ~ 0,
    BPQ040A == 2 ~ 0,
    BPQ050A == 1 ~ 1,
    TRUE ~ NA )) %>%
  dplyr::select(SEQN, BPMEDS)
tchol_2017 <- nhanes("TCHOL_J") %>%
  dplyr::select(SEQN, LBXTC) %>%
  rename(TOTCHOL = LBXTC)
hdl_2017 <- nhanes("HDL_J") %>%
```

```
  dplyr::select(SEQN, LBDHDD) %>%
  rename(HDLC = LBDHDD)
diq_2017 <- nhanes("DIQ_J") %>%
  mutate(DIABETES = case_when(DIQ010 == 1 ~ 1,
                              DIQ010 %in% c(2,3) ~ 0,
                              TRUE ~ NA)) %>%
  dplyr::select(SEQN, DIABETES)

# Join data from different tables
df_2017 <- bpx_2017 %>%
  full_join(demo_2017, by = "SEQN") %>%
  full_join(bmx_2017, by = "SEQN") %>%
  full_join(hdl_2017, by = "SEQN") %>%
  full_join(smq_2017, by = "SEQN") %>%
  full_join(bpq_2017, by = "SEQN") %>%
  full_join(tchol_2017, by = "SEQN") %>%
  full_join(diq_2017, by = "SEQN")

#Factor variables
df_2017 <- df_2017 %>%
  mutate(SEX = as.factor(SEX),
         CURSMOKE = as.factor(CURSMOKE),
         DIABETES = as.factor(DIABETES),
         BPMEDS = as.factor(BPMEDS)
  )

# Get blood pressure based on whether or not on BPMEDS
df_2017$SYSBP_UT <- ifelse(df_2017$BPMEDS == 0,
                           df_2017$SYSBP, 0)
df_2017$SYSBP_T <- ifelse(df_2017$BPMEDS == 1,
                          df_2017$SYSBP, 0)

#Eligibility Criteria: Age between 31-81
df_2017 <- df_2017 %>% filter(AGE >= 30 & AGE <=81 )
```

```
#MISSING DATA
vis_miss(df_2017 %>% dplyr::select(HDLC,TOTCHOL,AGE,BPMEDS,SYSBP_UT,SYSBP_T,
    CURSMOKE,DIABETES))

apply(df_2017,2,function(x) sum(is.na(x))/nrow(df_2017))
sum(complete.cases(df_2017))/nrow(df_2017)

#Complete case Analysis
```

```r
df_2017 <- na.omit(df_2017)
```

```r
#Calulate Brier score for two models in framingham dataset.
#Men Brierscore
pred_prob_men <- predict(mod_men, framingham_df_men, type = "response")
brier_men_fram <- sum((framingham_df_men$CVD - pred_prob_men)^2)/nrow(
    framingham_df_men)

#Women Brierscore
pred_prob_women <- predict(mod_women, framingham_df_women, type = "response")
brier_women_fram <- sum((framingham_df_women$CVD - pred_prob_women)^2)/nrow(
    framingham_df_women)

c(brier_men_fram,brier_women_fram)
```

```r
#Merge datasets while select only variables in model
#The variable S is indicator for individual being on the framingham dataset
fram_df_men <- framingham_df_men %>%
  dplyr::select(CVD,HDLC,TOTCHOL,AGE,SYSBP_UT,SYSBP_T,CURSMOKE,DIABETES)
y_men <- framingham_df_men %>% dplyr::select(CVD)
fram_df_men$S <- 1

fram_df_women <- framingham_df_women %>%
  dplyr::select(CVD,HDLC,TOTCHOL,AGE,SYSBP_UT,SYSBP_T,CURSMOKE,DIABETES)
y_women <- framingham_df_women %>% dplyr::select(CVD)
fram_df_women$S <- 1

df_2017 <- df_2017 %>%
  dplyr::select(HDLC,TOTCHOL,AGE,SYSBP_UT,SYSBP_T,CURSMOKE,DIABETES,SEX)
df_2017$S <- 0
df_2017$CVD <- NA

#re-order variables
df_2017 <- df_2017 %>%
  dplyr::select(CVD,HDLC,TOTCHOL,AGE,SYSBP_UT,SYSBP_T,CURSMOKE,DIABETES,SEX,S)


nhanes_men <- df_2017 %>%
  filter(SEX == "1") %>%
  dplyr::select(!SEX)

nhanes_women <- df_2017 %>%
  filter(SEX == "2") %>%
```

```
  dplyr::select(!SEX)

#merge data
join_men <- rbind(fram_df_men,nhanes_men)
join_women <- rbind(fram_df_women, nhanes_women)

#factor S
join_men$S <- as.factor(join_men$S)
join_women$S <- as.factor(join_women$S)

#
    ##############################################################################

####MEN
#FIT Inverse Odds
inv_odd_men <- glm(as.factor(S) ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT
    +1)+
                       log(SYSBP_T+1)+CURSMOKE+DIABETES,
                     data = join_men, family = "binomial")
#predict
prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
#weights
inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)
#attach to dataset
join_men$pred_prob_weighted <- inv_prob_fram_men
join_men$pred_prob <- prob_fram_men

#calculate estimate of brier score
brier_men_nhanes_2 <- join_men %>%
  filter(S == 1) %>%
  summarise(brier_num = sum(pred_prob_weighted * (pred_prob - CVD)^2)) %>%
  pull(brier_num) / join_men %>% filter(S==0) %>% count()

#another way to calculate (gives the same)
nhanes_men_brier <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
    prob_fram_men[1:length(y_men$CVD)])^2)/nrow(nhanes_men)

####WOMEN
#FIT Inverse Odds
#Do the same:
inv_odd_women <- glm(as.factor(S) ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_
    UT+1)+
                         log(SYSBP_T+1)+CURSMOKE+DIABETES ,data = join_women,
```

```
                            family = "binomial")
prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)
join_women$pred_prob_weighted <- inv_prob_fram_women
join_women$pred_prob <- prob_fram_women



nhanes_women_brier <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_women$
    CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(nhanes_women)

brier_women_nhanes_2 <- join_women %>%
  filter(S == 1) %>%
  summarise(brier_num = sum(pred_prob_weighted * (pred_prob - CVD)^2)) %>%
  pull(brier_num) / join_women %>% filter(S==0) %>% count()


#both estimates of brier scores
c(nhanes_men_brier,nhanes_women_brier)
c(brier_men_nhanes_2,brier_women_nhanes_2)
```

```
#MODEL USES: TOTCHOL,AGE,SYSBP,DIABP,HLDC SYSBP_T, SYSBP_UT, CURSMOKE and
    DIABETES
#SYSBP_UT and SYSBP_T are based on categorical variable BPMEDS

#Here we plot distributions of numeric variables of framinham dataset
#Continuous: TOTCHOL, AGE, DIABP, HDLC SYSSBP_T SYSBP_UT

##HLDC
hdlc_women <- ggplot(fram_df_women, aes(x = log(HDLC)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$HDLC)),
      sd = sd(log(fram_df_women$HDLC))), color = "red")+
  theme_minimal() +
  ggtitle("Log(HDLC)")
hdlc_men <- ggplot(fram_df_men, aes(x = log(HDLC)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$HDLC)),
     sd = sd(log(fram_df_men$HDLC))), color = "red")+
  theme_minimal() +
  ggtitle("Log(HDLC)")


##TOTCHOL
totchol_women <- ggplot(fram_df_women, aes(x = log(TOTCHOL)))+geom_density()+
```

```
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$TOTCHOL
    )), sd = sd(log(fram_df_women$TOTCHOL))), color = "red")+
  theme_minimal() +
  ggtitle("Log(TOTCHOL)")
totchol_men <- ggplot(fram_df_men, aes(x = log(TOTCHOL)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$TOTCHOL))
    , sd = sd(log(fram_df_men$TOTCHOL))), color = "red")+
  theme_minimal() +
  ggtitle("Log(TOTCHOL)")

##AGE
age_women <- ggplot(fram_df_women, aes(x = log(AGE)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$AGE)),
    sd = sd(log(fram_df_women$AGE))), color = "red")+
  theme_minimal() +
  ggtitle("Log(AGE)")
age_men <- ggplot(fram_df_men, aes(x = log(AGE)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$AGE)), sd
     = sd(log(fram_df_men$AGE))), color = "red")+
  theme_minimal() +
  ggtitle("Log(AGE)")



sys_women <- ggplot(fram_df_women, aes(x = log(AGE)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$AGE)),
    sd = sd(log(fram_df_women$AGE))), color = "red")+
  theme_minimal() +
  ggtitle("Log(AGE)")

##SYSBP
xx <- fram_df_women %>% dplyr::select(SYSBP_T) %>% filter(SYSBP_T != 0)
xx <- as.data.frame(xx)
sysbpT_women <- ggplot(xx,aes(log(SYSBP_T)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_T)), sd = sd
    (log(xx$SYSBP_T))), color = "red")+
  theme_minimal() +
  ggtitle("Log(SYSBP_T)")

xx <- fram_df_men %>% dplyr::select(SYSBP_T) %>% filter(SYSBP_T != 0)
xx <- as.data.frame(xx)
sysbpT_men <- ggplot(xx,aes(log(SYSBP_T)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_T)), sd = sd
```

```
      (log(xx$SYSBP_T))), color = "red")+
  theme_minimal() +
  ggtitle("Log(SYSBP_T)")

##SYSBP_UT
xx <- fram_df_women %>% dplyr::select(SYSBP_UT) %>% filter(SYSBP_UT != 0)
xx <- as.data.frame(xx)
sysbpUT_women <- ggplot(xx,aes(log(SYSBP_UT)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_UT)), sd =
      sd(log(xx$SYSBP_UT))), color = "red")+
  theme_minimal() +
  ggtitle("Log(SYSBP_UT)")

xx <- fram_df_men %>% dplyr::select(SYSBP_UT) %>% filter(SYSBP_UT != 0)
xx <- as.data.frame(xx)
sysbpUT_men <- ggplot(xx,aes(log(SYSBP_UT)))+geom_density()+
  stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_UT)), sd =
      sd(log(xx$SYSBP_UT))), color = "red")+
  theme_minimal() +
  ggtitle("Log(SYSBP_UT)")

p1 <- ggarrange(hdlc_men,totchol_men,age_men,sysbpT_men,sysbpUT_men,nrow = 5)
    %>% annotate_figure(top = text_grob("Women"))
p2 <- ggarrange(hdlc_women,totchol_women,age_women,sysbpT_women,sysbpUT_women,
    nrow=5) %>% annotate_figure(top = text_grob("Men"))

ggarrange(p1,p2,ncol = 2) %>% annotate_figure(top = text_grob("Continuos
    Densities with Normal Approximations", size = 14, face = "bold"))
```

```
#
    ##############################################################################################


#FUNCTION TO GENERATE WHILE VARYING CORRELATIONS WOMEN

sim_data_function_men <- function(data,mu,cov){

framingham_df_men_log <- data %>%
  mutate(log_HDLC = log(HDLC),
         log_TOTCHOL = log(TOTCHOL),
         log_AGE = log(AGE),
         log_SYSBP = log(SYSBP)) %>%
  dplyr::select(log_HDLC,log_TOTCHOL,log_AGE,log_SYSBP,BPMEDS,CURSMOKE,
```

```
      DIABETES)

sim_data <- mvrnorm(n = 1000, mu = mu, Sigma = cov)
sim_data <- as.data.frame(sim_data)

#CHANGE BPMEDS TO CATEGORICAL
pp <- sum(as.numeric(data$BPMEDS)-1)/nrow(data)
sim_data$BPMEDS <- ifelse(sim_data$BPMEDS > quantile(sim_data$BPMEDS, 1 - pp),
    1, 0)

#CHANGE DIABETES TO CATEGORICAL
pp <- sum(as.numeric(data$DIABETES)-1)/nrow(data)
sim_data$DIABETES <- ifelse(sim_data$DIABETES > quantile(sim_data$DIABETES, 1
   - pp), 1, 0)

#CHANGE CURSMOKE TO CATEGORICAL
pp <- sum(as.numeric(data$CURSMOKE)-1)/nrow(data)
sim_data$CURSMOKE <- ifelse(sim_data$CURSMOKE > quantile(sim_data$CURSMOKE, 1
   - pp), 1, 0)


return(sim_data)
}


#FUNCTION TO GENERATE WHILE VARYING CORRELATIONS WOMEN

sim_data_function_women <- function(data,mu,cov){

  framingham_df_women_log <- data %>%
  mutate(log_HDLC = log(HDLC),
         log_TOTCHOL = log(TOTCHOL),
         log_AGE = log(AGE),
         log_SYSBP = log(SYSBP)) %>%
  dplyr::select(log_HDLC,log_TOTCHOL,log_AGE,log_SYSBP,BPMEDS,CURSMOKE,
      DIABETES)

sim_data <- mvrnorm(n = 1000, mu = mu, Sigma = cov)
sim_data <- as.data.frame(sim_data)

#CHANGE BPMEDS TO CATEGORICAL
pp <- sum(as.numeric(data$BPMEDS)-1)/nrow(data)
```

```r
sim_data$BPMEDS <- ifelse(sim_data$BPMEDS > quantile(sim_data$BPMEDS, 1 - pp), 1, 0)

#CHANGE DIABETES TO CATEGORICAL
pp <- sum(as.numeric(data$DIABETES)-1)/nrow(data)
sim_data$DIABETES <- ifelse(sim_data$DIABETES > quantile(sim_data$DIABETES, 1 - pp), 1, 0)

#CHANGE CURSMOKE TO CATEGORICAL
pp <- sum(as.numeric(data$CURSMOKE)-1)/nrow(data)
sim_data$CURSMOKE <- ifelse(sim_data$CURSMOKE > quantile(sim_data$CURSMOKE, 1 - pp), 1, 0)

return(sim_data)
}
```

```r
##HIGLY CORRELATED DATASET
#https://stats.stackexchange.com/questions/124538/how-to-generate-a-large-full-rank-random-correlation-matrix-with-some-strong-cor
# Generate W matrix with random values from a standard normal distribution

v <- 7          # 12 variables
f <- 5          # Subset-correlation
vg <- v / f     # Variables per subset
a <- 5
d <- 100
# Constructing a factor matrix 'L' with higher magnitude positive and negative relationships
set.seed(3)
L <- matrix(c(
  runif(vg*f, -a, a), runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d,
  runif(vg*f, -a, a)/d, runif(vg*f, -a, a), runif(vg*f, -a, a)/d,
  runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d, runif(vg*f, -a, a)
), nrow = v, ncol = v)

names <- c("log_HDLC","log_TOTCHOL","log_AGE","log_SYSBP","BPMEDS","CURSMOKE","DIABETES")
colnames(L) <- names
rownames(L) <- names

# Make covariance and correlation matrix
cov_matrix <- L %*% t(L)
colnames(cov_matrix) <- names
```

```r
rownames(cov_matrix) <- names # L multiplied with its transpose
cor_high<- cov2cor(cov_matrix)
```

```r
##Medium CORRELATED DATASET
#https://stats.stackexchange.com/questions/124538/how-to-generate-a-large-full-rank-random-correlation-matrix-with-some-strong-cor
# Generate W matrix with random values from a standard normal distribution

v <- 7           # 12 variables
f <- 1           # Subset-correlation based on 5 common factors
vg <- v / f      # Variables per subset
a <- .1
d <- 5

# Constructing a factor matrix 'L' with higher magnitude positive and negative relationships
set.seed(1)
L <- matrix(c(
  runif(vg*f, -a, a), runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d,
  runif(vg*f, -a, a)/d, runif(vg*f, -a, a), runif(vg*f, -a, a)/d,
  runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d, runif(vg*f, -a, a)
), nrow = v, ncol = v)


colnames(L) <- names
rownames(L) <- names

# Make covariance and correlation matrix
cov_matrix <- L %*% t(L)
colnames(cov_matrix) <- names
rownames(cov_matrix) <- names # L multiplied with its transpose
cor_med <- cov2cor(cov_matrix)

ggcorrplot(cor_med)
```

```r
#Statistic from dataset

#MEN
framingham_df_men_log <- framingham_df_men %>%
  mutate(log_HDLC = log(HDLC),
         log_TOTCHOL = log(TOTCHOL),
         log_AGE = log(AGE),
         log_SYSBP = log(SYSBP)) %>%
```

```
  dplyr::select(log_HDLC,log_TOTCHOL,log_AGE,log_SYSBP,BPMEDS,CURSMOKE,
      DIABETES)

mu_men <- framingham_df_men_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  summarize_if(is.numeric,mean)

sd_men <- framingham_df_men_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  summarize_if(is.numeric,sd)

#WOMEN
framingham_df_women_log <- framingham_df_women %>%
  mutate(log_HDLC = log(HDLC),
         log_TOTCHOL = log(TOTCHOL),
         log_AGE = log(AGE),
         log_SYSBP = log(SYSBP)) %>%
  dplyr::select(log_HDLC,log_TOTCHOL,log_AGE,log_SYSBP,BPMEDS,CURSMOKE,
      DIABETES)

mu_women <- framingham_df_women_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  summarize_if(is.numeric,mean)

sd_women <- framingham_df_women_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  summarize_if(is.numeric,sd)



#NO CORRELATION
cor_no <- diag(1,7)
colnames(cor_no) <- names
rownames(cor_no) <- names

```

```r
cov_no <- diag(sd_men) %*% cor_no %*% diag(sd_men)
colnames(cov_no) <- names
rownames(cov_no) <- names

cov_no_women <- diag(sd_women) %*% cor_no %*% diag(sd_women)
colnames(cov_no_women) <- names
rownames(cov_no_women) <- names
```

```r
p.cor.high <- ggcorrplot(cor_high) + ggtitle("High Correlation")
p.cor.med <- ggcorrplot(cor_med) + ggtitle("Medium Correlation")
p.cor.no <- ggcorrplot(cor_no) + ggtitle("No Correlation")

ggarrange(p.cor.no,p.cor.med,p.cor.high,common.legend = T)

ggarrange(ggarrange(p.cor.no,p.cor.med,ncol = 2, legend = "none"),p.cor.high,
    nrow = 2, legend = "bottom")

###################################
#SIMILAR CORRELATION WITH FRAMINGHAM DATASET
cov_men <- framingham_df_men_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cov()


sim_data_similar_men <- sim_data_function_men(framingham_df_men, as.numeric(mu
    _men), cov_men)

# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_similar_men$log_SYSBP_UT <- ifelse(sim_data_similar_men$BPMEDS == 0,
                            sim_data_similar_men$log_SYSBP, 0)
sim_data_similar_men$log_SYSBP_T <- ifelse(sim_data_similar_men$BPMEDS == 1,
                           sim_data_similar_men$log_SYSBP, 0)

sim_data_similar_men_2 <- sim_data_similar_men %>%
  dplyr::select(log_HDLC,log_TOTCHOL,log_AGE, log_SYSBP_UT,log_SYSBP_T,
      CURSMOKE,DIABETES)

p1 <- sim_data_similar_men %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("MEN Framingham Simulated Correlation")
```

```r

p <- framingham_df_men_log %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cor() %>%
  ggcorrplot()+ggtitle("Framingham Correlation") #same summary statistics


##SIMULATION
brier_similar_men <- numeric(0)
for(i in 1:5000){
sim_data_similar_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_men), cov_men)

# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_similar_men$log_SYSBP_UT <- ifelse(sim_data_similar_men$BPMEDS == 0,
                           sim_data_similar_men$log_SYSBP, 0)
sim_data_similar_men$log_SYSBP_T <- ifelse(sim_data_similar_men$BPMEDS == 1,
                          sim_data_similar_men$log_SYSBP, 0)

##ESTIMATE BRIER SCORE
####MEN
framingham_df_men_log$S <- 1
#FIT Inverse Odds
framingham_df_men_log$log_SYSBP_UT <- ifelse(framingham_df_men_log$BPMEDS == 0,
                           framingham_df_men_log$log_SYSBP, 0)
framingham_df_men_log$log_SYSBP_T <- ifelse(framingham_df_men_log$BPMEDS == 1,
                          framingham_df_men_log$log_SYSBP, 0)

sim_data_similar_men$S <- 0
join_men <- rbind(framingham_df_men_log,sim_data_similar_men)

inv_odd_men <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_SYSBP_UT
                   +log_SYSBP_T + CURSMOKE+DIABETES,
                   data = join_men, family = "binomial")
prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)


brier_similar_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD
```

```
    - prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_similar_men)
print(i)
}
brier_similar_men_avg <- mean(brier_similar_men)
brier_similar_men_sd <- sd(brier_similar_men)
pbrier_similar_men <- ggplot(as.data.frame(brier_similar_men),aes(x = brier_
    similar_men))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Similar Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()
```

```
############GENERATE DATA

##HIGH CORRELATION
cov_men <- diag(sd_men) %*% cor_high %*% diag(sd_men)
colnames(cov_men) <- names
rownames(cov_men) <- names

sim_data_high_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
    men), cov_men)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_high_men$log_SYSBP_UT <- ifelse(sim_data_high_men$BPMEDS == 0,
                            sim_data_high_men$log_SYSBP, 0)
sim_data_high_men$log_SYSBP_T <- ifelse(sim_data_high_men$BPMEDS == 1,
                           sim_data_high_men$log_SYSBP, 0)

p2 <- sim_data_high_men %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("High Correlation")

##SIMULATION
brier_high_men <- numeric(0)
for(i in 1:5000){
sim_data_high_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
    men), cov_men)

```

```r


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_high_men$log_SYSBP_UT <- ifelse(sim_data_high_men$BPMEDS == 0,
                            sim_data_high_men$log_SYSBP, 0)
sim_data_high_men$log_SYSBP_T <- ifelse(sim_data_high_men$BPMEDS == 1,
                           sim_data_high_men$log_SYSBP, 0)


##ESTIMATE BRIER SCORE
####MEN
#FIT Inverse Odds
framingham_df_men_log$S <- 1
sim_data_high_men$S <- 0

join_men <- rbind(framingham_df_men_log, sim_data_high_men)
inv_odd_men <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_SYSBP
    _UT
                   +log_SYSBP_T + CURSMOKE+DIABETES,
                   data = join_men, family = "binomial")
prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)


brier_high_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
    prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_high_men)
print(i)
}

brier_high_men_avg <- mean(brier_high_men)
brier_high_men_sd <- sd(brier_high_men)
pbrier_high_men <- ggplot(as.data.frame(brier_high_men),aes(x = brier_high_men
    ))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "High Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()

##################
#Medium
```

```r
##GENERATE DATA

cov_men <- diag(sd_men) %*% cor_med %*% diag(sd_men)
colnames(cov_men) <- names
rownames(cov_men) <- names

sim_data_med_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_men
    ), cov_men)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_med_men$log_SYSBP_UT <- ifelse(sim_data_med_men$BPMEDS == 0,
                            sim_data_med_men$log_SYSBP, 0)
sim_data_med_men$log_SYSBP_T <- ifelse(sim_data_med_men$BPMEDS == 1,
                           sim_data_med_men$log_SYSBP, 0)

p3 <- sim_data_med_men %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("Medium Correlation")

#SIMULATION
brier_med_men <- numeric(0)
for(i in 1:5000){
sim_data_med_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_men
    ), cov_men)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_med_men$log_SYSBP_UT <- ifelse(sim_data_med_men$BPMEDS == 0,
                            sim_data_med_men$log_SYSBP, 0)
sim_data_med_men$log_SYSBP_T <- ifelse(sim_data_med_men$BPMEDS == 1,
                           sim_data_med_men$log_SYSBP, 0)
##ESTIMATE BRIER SCORE
####MEN
#FIT Inverse Odds
sim_data_med_men$S <- 0



join_men <- rbind(framingham_df_men_log,sim_data_med_men)
inv_odd_men <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_SYSBP
    _UT
```

```
                    +log_SYSBP_T + CURSMOKE+ DIABETES,
                    data = join_men, family = "binomial")
prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)


brier_med_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
    prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_med_men)
print(i)
}

brier_med_men_avg <- mean(brier_med_men)
brier_med_men_sd <- sd(brier_med_men)
pbrier_med_men <- ggplot(as.data.frame(brier_med_men),aes(x = brier_med_men))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Medium Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()

##NO CORR
sim_data_nocor_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
    men), cov_no)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_nocor_men$log_SYSBP_UT <- ifelse(sim_data_nocor_men$BPMEDS == 0,
                            sim_data_nocor_men$log_SYSBP, 0)
sim_data_nocor_men$log_SYSBP_T <- ifelse(sim_data_nocor_men$BPMEDS == 1,
                           sim_data_nocor_men$log_SYSBP, 0)

p4 <- sim_data_nocor_men %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("No Correlation (Men)")

#SIMULATION
brier_nocor_men <- numeric(0)
for(i in 1:5000){
sim_data_nocor_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
    men), cov_no)
```

```r


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_nocor_men$log_SYSBP_UT <- ifelse(sim_data_nocor_men$BPMEDS == 0,
                            sim_data_nocor_men$log_SYSBP, 0)
sim_data_nocor_men$log_SYSBP_T <- ifelse(sim_data_nocor_men$BPMEDS == 1,
                           sim_data_nocor_men$log_SYSBP, 0)
##ESTIMATE BRIER SCORE
####MEN
#FIT Inverse Odds
sim_data_nocor_men$S <- 0

join_men <- rbind(framingham_df_men_log,sim_data_nocor_men)
inv_odd_men <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_SYSBP
    _UT
                   +log_SYSBP_T + CURSMOKE+ DIABETES,
                   data = join_men, family = "binomial")
prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)


brier_nocor_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
    prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_nocor_men)
print(i)
}

brier_nocor_men_avg <- mean(brier_nocor_men)
brier_nocor_men_sd <- sd(brier_nocor_men)
pbrier_nocor_men <- ggplot(as.data.frame(brier_nocor_men),aes(x = brier_nocor_
    men))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Independent Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()
```

```r
  #
      ##############################################################################
      WOMEN
      ##############################################################################
```

```r


###################################
#SIMILAR CORRELATION WITH FRAMINGHAM DATASET
cov_women <- framingham_df_women_log %>%
  dplyr::select(names) %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cov()


sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
    numeric(mu_women), cov_women)

# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS ==
     0,
                                              sim_data_similar_women$log_SYSBP,
                                                 0)
sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS ==
    1,
                                            sim_data_similar_women$log_SYSBP,
                                               0)


p1 <- sim_data_similar_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("women Framingham Simulated Correlation")

p <- framingham_df_women_log %>%
  dplyr::select(-S)%>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cor() %>%
  ggcorrplot()+ggtitle("Framingham Correlation") #same summary statistics


##SIMULATION
brier_similar_women <- numeric(0)
```

```r
for(i in 1:5000){
  sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
      numeric(mu_women), cov_women)

  # Get blood pressure based on whether or not on BPMEDS on simulated data
  sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS
      == 0,
                                                sim_data_similar_women$log_SYSBP
                                                    , 0)
  sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS
      == 1,
                                               sim_data_similar_women$log_SYSBP,
                                                   0)



  ##ESTIMATE BRIER SCORE
  ####women
  #FIT Inverse Odds
  framingham_df_women_log$log_SYSBP_UT <- ifelse(framingham_df_women_log$
      BPMEDS == 0,
                                                 framingham_df_women_log$log_
                                                    SYSBP, 0)
  framingham_df_women_log$log_SYSBP_T <- ifelse(framingham_df_women_log$BPMEDS
       == 1,
                                                framingham_df_women_log$log_
                                                   SYSBP, 0)
framingham_df_women_log$S <- 1
sim_data_similar_women$S <- 0

join_women <- rbind(framingham_df_women_log,sim_data_similar_women)

inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE +
                       log_SYSBP_UT+log_SYSBP_T + CURSMOKE+DIABETES,
                     data = join_women, family = "binomial")
  prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
  inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)


  brier_similar_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
      women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
      similar_women)
  print(i)

```

```
}
brier_similar_women_avg <- mean(brier_similar_women)
brier_similar_women_sd <- sd(brier_similar_women)
pbrier_similar_women <- ggplot(as.data.frame(brier_similar_women),aes(x =
    brier_similar_women))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Similar Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()




###################################
#SIMILAR CORRELATION WITH FRAMINGHAM DATASET
cov_women <- framingham_df_women_log %>%
  dplyr::select(names) %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cov()


sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
    numeric(mu_women), cov_women)

# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS ==
     0,
                                              sim_data_similar_women$log_SYSBP,
                                                0)
sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS ==
    1,
                                            sim_data_similar_women$log_SYSBP,
                                              0)

p1 <- sim_data_similar_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("women Framingham Simulated Correlation")
```

```
108
109 p <- framingham_df_women_log %>%
110   dplyr::select(-S) %>%
111   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
112          DIABETES = as.numeric(DIABETES)-1,
113          BPMEDS = as.numeric(BPMEDS)-1) %>%
114   cor() %>%
115   ggcorrplot()+ggtitle("Framingham Correlation") #same summary statistics
116
117
118 ##SIMULATION
119 brier_similar_women <- numeric(0)
120 for(i in 1:5000){
121   sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
        numeric(mu_women), cov_women)
122
123   # Get blood pressure based on whether or not on BPMEDS on simulated data
124   sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS
        == 0,
125                                                 sim_data_similar_women$log_SYSBP
                                                      , 0)
126   sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS
        == 1,
127                                                sim_data_similar_women$log_SYSBP,
                                                      0)
128
129   ##ESTIMATE BRIER SCORE
130   ####women
131   #FIT Inverse Odds
132   framingham_df_women_log$log_SYSBP_UT <- ifelse(framingham_df_women_log$
        BPMEDS == 0,
133                                                  framingham_df_women_log$log_
                                                       SYSBP, 0)
134   framingham_df_women_log$log_SYSBP_T <- ifelse(framingham_df_women_log$BPMEDS
         == 1,
135                                                 framingham_df_women_log$log_
                                                      SYSBP, 0)
136   framingham_df_women_log$S <- 1
137   sim_data_similar_women$S <- 0
138
139   join_women <- rbind(framingham_df_women_log,sim_data_similar_women)
140
141   inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
```

```
      SYSBP_UT
                        +log_SYSBP_T + CURSMOKE+DIABETES,
                        data = join_women, family = "binomial")
  prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
  inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)


  brier_similar_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
      women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
      similar_women)
  print(i)
}
brier_similar_women_avg <- mean(brier_similar_women)
brier_similar_women_sd <- sd(brier_similar_women)
pbrier_similar_women <- ggplot(as.data.frame(brier_similar_women),aes(x =
    brier_similar_women))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Similar Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()
```

```
##HIGH CORRELATION
cov_women <- diag(sd_women) %*% cor_high %*% diag(sd_women)
colnames(cov_women) <- names
rownames(cov_women) <- names

sim_data_high_women <- sim_data_function_women(framingham_df_women, as.numeric
    (mu_women), cov_women)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_high_women$log_SYSBP_UT <- ifelse(sim_data_high_women$BPMEDS == 0,
                                           sim_data_high_women$log_SYSBP, 0)
sim_data_high_women$log_SYSBP_T <- ifelse(sim_data_high_women$BPMEDS == 1,
                                          sim_data_high_women$log_SYSBP, 0)

p2 <- sim_data_high_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("High Correlation")
```

```r

##SIMULATION
brier_high_women <- numeric(0)
for(i in 1:5000){
  sim_data_high_women <- sim_data_function_women(framingham_df_women, as.
     numeric(mu_women), cov_women)


  # Get blood pressure based on whether or not on BPMEDS on simulated data
  sim_data_high_women$log_SYSBP_UT <- ifelse(sim_data_high_women$BPMEDS == 0,
                                             sim_data_high_women$log_SYSBP, 0)
  sim_data_high_women$log_SYSBP_T <- ifelse(sim_data_high_women$BPMEDS == 1,
                                            sim_data_high_women$log_SYSBP, 0)
  ##ESTIMATE BRIER SCORE
  ####women
  #FIT Inverse Odds
  sim_data_high_women$S <- 0

  join_women <- rbind(framingham_df_women_log,sim_data_high_women)
  inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
     SYSBP_UT
                       +log_SYSBP_T + CURSMOKE+ DIABETES,
                       data = join_women, family = "binomial")
  prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
  inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)


  brier_high_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
     women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
     high_women)
  print(i)
}

brier_high_women_avg <- mean(brier_high_women)
brier_high_women_sd <- sd(brier_high_women)
pbrier_high_women <- ggplot(as.data.frame(brier_high_women),aes(x = brier_high
    _women))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
     width
  labs(
    title = "High Correlation",
    x = "Brier Score",
    y = "Frequency")+
```

```r
  theme_minimal()

#######
#Medium
##GENERATE DATA

cov_women <- diag(sd_women) %*% cor_med %*% diag(sd_women)
colnames(cov_women) <- names
rownames(cov_women) <- names

sim_data_med_women <- sim_data_function_women(framingham_df_women, as.numeric(
    mu_women), cov_women)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_med_women$log_SYSBP_UT <- ifelse(sim_data_med_women$BPMEDS == 0,
                                          sim_data_med_women$log_SYSBP, 0)
sim_data_med_women$log_SYSBP_T <- ifelse(sim_data_med_women$BPMEDS == 1,
                                         sim_data_med_women$log_SYSBP, 0)

p3 <- sim_data_med_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("Medium Correlation")

#SIMULATION
brier_med_women <- numeric(0)
for(i in 1:5000){
  sim_data_med_women <- sim_data_function_women(framingham_df_women, as.
      numeric(mu_women), cov_women)


  # Get blood pressure based on whether or not on BPMEDS on simulated data
  sim_data_med_women$log_SYSBP_UT <- ifelse(sim_data_med_women$BPMEDS == 0,
                                            sim_data_med_women$log_SYSBP, 0)
  sim_data_med_women$log_SYSBP_T <- ifelse(sim_data_med_women$BPMEDS == 1,
                                           sim_data_med_women$log_SYSBP, 0)
  ##ESTIMATE BRIER SCORE
  ####women
  #FIT Inverse Odds
  sim_data_med_women$S <- 0

  join_women <- rbind(framingham_df_women_log,sim_data_med_women)
```

```r
  inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
      SYSBP_UT
                       +log_SYSBP_T + CURSMOKE+ DIABETES,
                       data = join_women, family = "binomial")
  prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
  inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)


  brier_med_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
      women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_med
      _women)
  print(i)
}

brier_med_women_avg <- mean(brier_med_women)
brier_med_women_sd <- sd(brier_med_women)
pbrier_med_women <- ggplot(as.data.frame(brier_med_women),aes(x = brier_med_
    women))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Medium Correlation",
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()


######
##############
##NO CORR
sim_data_nocor_women <- sim_data_function_women(framingham_df_women, as.
    numeric(mu_women), cov_no)


# Get blood pressure based on whether or not on BPMEDS on simulated data
sim_data_nocor_women$log_SYSBP_UT <- ifelse(sim_data_nocor_women$BPMEDS == 0,
                                            sim_data_nocor_women$log_SYSBP, 0)
sim_data_nocor_women$log_SYSBP_T <- ifelse(sim_data_nocor_women$BPMEDS == 1,
                                           sim_data_nocor_women$log_SYSBP, 0)

p4 <- sim_data_nocor_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
  cor() %>%
```

```r
  ggcorrplot()+ ggtitle("No Correlation (women)")

#SIMULATION
brier_nocor_women <- numeric(0)
for(i in 1:5000){
  sim_data_nocor_women <- sim_data_function_women(framingham_df_women, as.
      numeric(mu_women), cov_no)


  # Get blood pressure based on whether or not on BPMEDS on simulated data
  sim_data_nocor_women$log_SYSBP_UT <- ifelse(sim_data_nocor_women$BPMEDS ==
      0,
                                              sim_data_nocor_women$log_SYSBP, 0)
  sim_data_nocor_women$log_SYSBP_T <- ifelse(sim_data_nocor_women$BPMEDS == 1,
                                             sim_data_nocor_women$log_SYSBP, 0)
  ##ESTIMATE BRIER SCORE
  ####women
  #FIT Inverse Odds
  sim_data_nocor_women$S <- 0

  join_women <- rbind(framingham_df_women_log,sim_data_nocor_women)
  inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
      SYSBP_UT
                       +log_SYSBP_T + CURSMOKE+ DIABETES,
                       data = join_women, family = "binomial")
  prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
  inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(-1)


  brier_nocor_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
      women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
      nocor_women)
  print(i)
}

brier_nocor_women_avg <- mean(brier_nocor_women)
brier_nocor_women_sd <- sd(brier_nocor_women)
pbrier_nocor_women <- ggplot(as.data.frame(brier_nocor_women),aes(x = brier_
    nocor_women))+
  geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
  labs(
    title = "Independent Correlation",
```

```
    x = "Brier Score",
    y = "Frequency")+
  theme_minimal()
```

```
p1 <- sim_data_similar_women %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T,-S) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("Women Framingham Simulated Correlation")

p <- framingham_df_women_log %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T,-S) %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cor() %>%
  ggcorrplot()+ggtitle("Women Framingham Correlation")



p2 <- sim_data_similar_men %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T,-S) %>%
  cor() %>%
  ggcorrplot()+ ggtitle("Men Framingham Simulated Correlation")

pp <- framingham_df_men_log %>%
  dplyr::select(-log_SYSBP_UT,-log_SYSBP_T,-S) %>%
  mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
         DIABETES = as.numeric(DIABETES)-1,
         BPMEDS = as.numeric(BPMEDS)-1) %>%
  cor() %>%
  ggcorrplot()+ggtitle("Men Framingham Correlation")



ggarrange(p1,p,p2,pp,ncol = 2, nrow = 2, common.legend = T, legend = "bottom")
```

```
#RESULTS
ggarrange(pbrier_nocor_men,pbrier_similar_men,pbrier_med_men,pbrier_high_men,
    ncol = 2,nrow = 2) %>% annotate_figure(top = text_grob("MEN Distribution
    of Estimate for Brier Score on Matrix", size = 14, face = "bold"))

res_men <- cbind(c(brier_nocor_men_avg, brier_similar_men_avg, brier_med_men_
    avg, brier_high_men_avg),c(brier_nocor_men_sd, brier_similar_men_sd, brier
    _med_men_sd, brier_high_men_sd))
```

```r
res_men <- round(res_men,4)
colnames(res_men) <- c("Brier", "SD Brier")
rownames(res_men) <- c("Uncorrelated", "Similar", "Medium", "High")
tab_men <- res_men %>%
  kable(booktabs = TRUE, caption = " Brier Results for Men Model") %>%
  kableExtra::kable_styling(font_size = 8, latex_options = "HOLD_position")
     %>%     add_footnote(paste("NHANES Brier score is", round(nhanes_men_
     brier,4)))

ggarrange(pbrier_nocor_women,pbrier_similar_women,pbrier_med_women,pbrier_high
    _women,ncol = 2,nrow = 2) %>% annotate_figure(top = text_grob("Women
    Distribution of Estimate for Brier Score on Matrix", size = 14, face = "
    bold"))

res_women <- cbind(c(brier_nocor_women_avg, brier_similar_women_avg, brier_med
    _women_avg, brier_high_women_avg),c(brier_nocor_women_sd, brier_similar_
    women_sd, brier_med_women_sd, brier_high_women_sd))
res_women <- round(res_women,4)
colnames(res_women) <- c("Brier", "SD Brier")
rownames(res_women) <- c("Uncorrelated", "Similar", "Medium", "High")
tab_women <- res_women %>%
  kable(booktabs = TRUE, caption = " Brier Results for Women Model") %>%
  kableExtra::kable_styling(font_size = 8, latex_options = "HOLD_position")
     %>%     add_footnote(paste("NHANES Brier score is", round(nhanes_women_
     brier,4)))

kable(list(tab_men, tab_women))
c(brier_men_fram,brier_women_fram)
tab_men
```