

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. Additionally, the study aims to replicate this evaluation in a simulated population, 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 appraise the efficacy of an established cardiovascular risk 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. Furthermore, the study aims to replicate this assessment within a simulated 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.

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].

2.3 Missing Data

In both the Framingham and NHANES datasets, missing data presents a challenge. For the scope of this study, considering the model's construction on the Framingham dataset comprising solely complete cases, a parallel approach of complete case analysis was adopted for the investigation of model transportability. Acknowledging the primary focus on transportation analysis, any selection bias resulting from the omission of missing values was disregarded for this specific study. The emphasis remained on assessing model performance across datasets, with the impact of missing data deemed secondary in light of the primary investigation into transportation dynamics.

3 Methodology

3.1 Model

Table 1 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 1: 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:

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

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 $\hat{\delta}(X)$ is an estimator for the inverse-odds weights in the test set given by

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

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 $\hat{\delta}(X)$.

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 aim of the simulation is 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 $n = 1000$ 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. 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 1. 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 a direct comparison between the estimated Brier scores and the original Brier score derived from the NHANES dataset by calculating its distance.

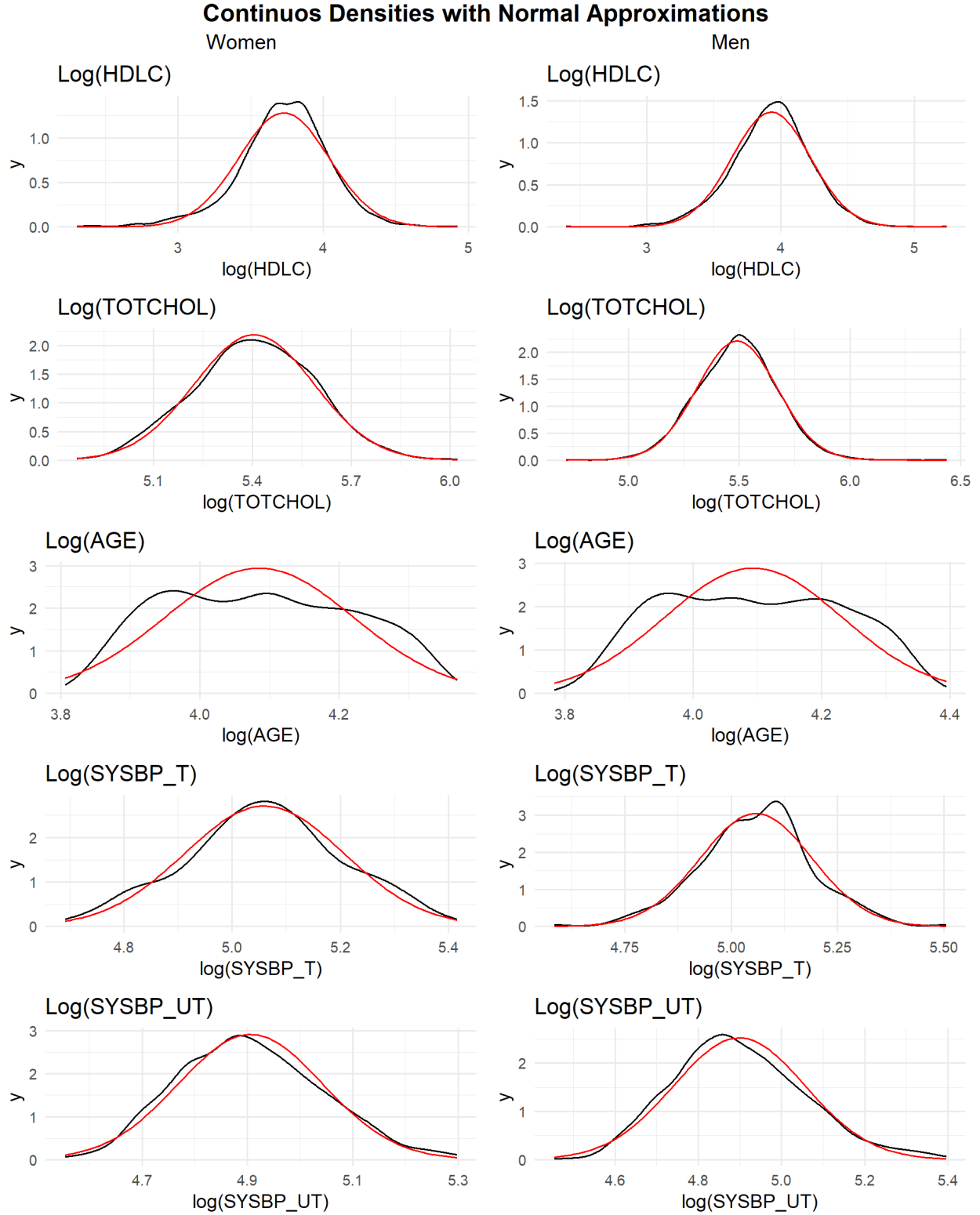


Figure 1: Approximations of log-transformed continuous variables using a normal distribution with mean and variance estimates derived from the logarithmically transformed variable

4 Results

The results for 5,000 simulations are presented in Figure 3 and in Table 2. Figure 3 illustrates the distributions concerning the estimated Brier scores across various simulation settings for both men and women. It's notable that all distributions exhibit a semblance of normality, characterized by low standard deviations. In Table 2, the mean and standard deviation of the estimate of the brier 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 0.1583 for men and 0.1402 for women. A noticeable discrepancy emerges in the NHANES Brier score for men, exhibiting 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 manifest notably higher values. Among women, these scores converge around 0.3, peaking notably in scenarios mirroring the correlation matrix of the Framingham dataset, while recording lower values for both uncorrelated and moderately correlated variables. Conversely, for males, the estimated scores hover around 0.25. In this context, both highly correlated structures and those resembling the correlation patterns of the Framingham dataset yield elevated estimated Brier scores, contrasting with the lower estimation observed for moderately correlated variables. Intriguingly, the scenario with uncorrelated variables also demonstrates an estimated Brier score close to the highest observed value. Both estimated brier score signifies a moderate level of inaccuracy or uncertainty in the model's predictions. This score indicates that the model's forecasted probabilities for certain events or outcomes exhibit notable discrepancies from the actual observed outcomes.

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 slightly less accurate or more uncertain. Conversely, when the simulated data exhibit moderately correlated variables, the estimated Brier scores are slightly lower. This implies that moderate correlation structures in the data contribute to a slightly more favorable predictive performance, indicating better calibration or accuracy of the model's predictions compared to highly correlated structures.

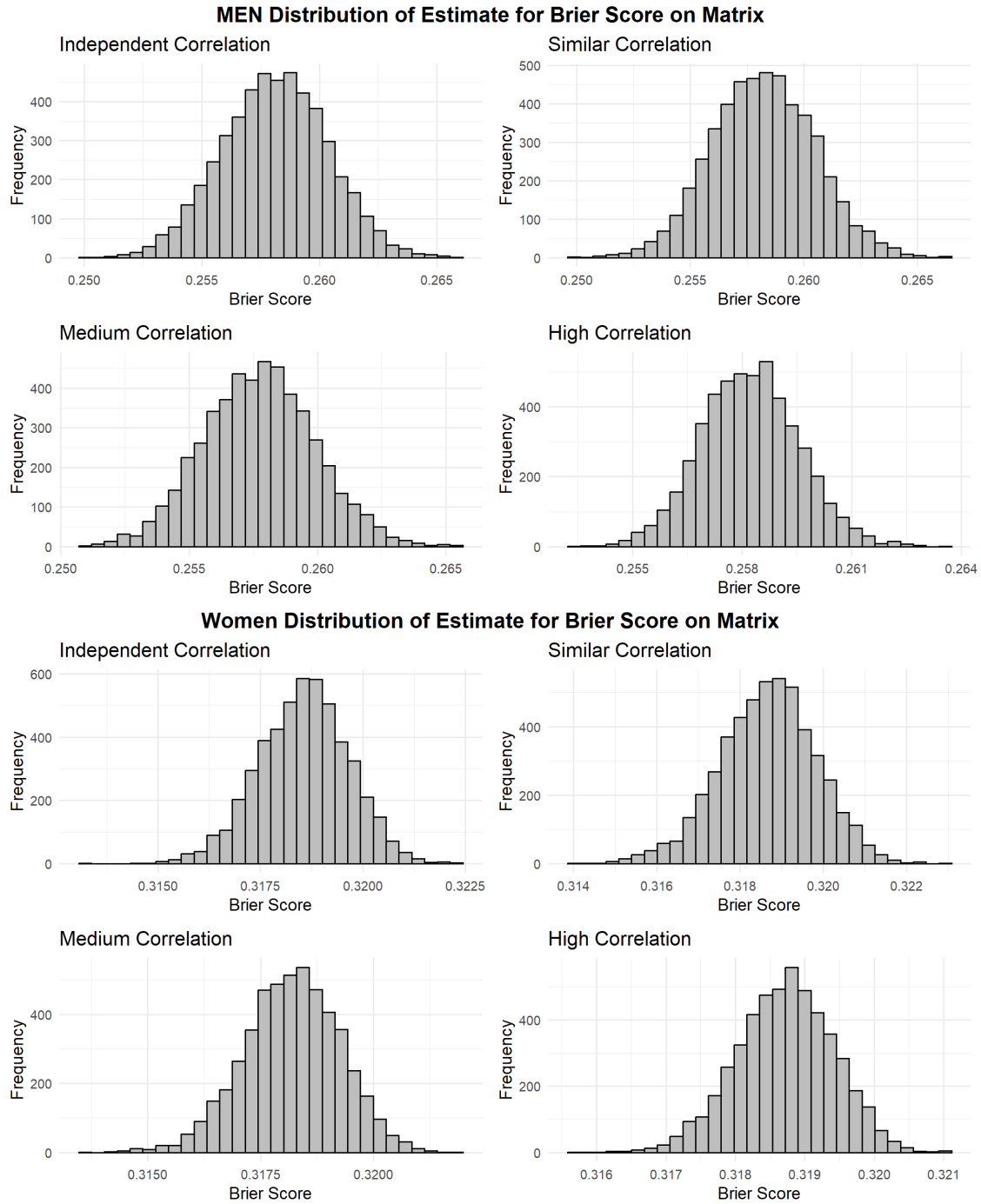


Figure 3: Distribution of the estimated brier score

Model	Estimate for Brier	SD Brier	Model	Estimate for Brier	SD Brier
Uncorrelated	0.3185	0.0011	Uncorrelated	0.2581	0.0023
Similar	0.3187	0.0012	Similar	0.2582	0.0023
Medium	0.3182	0.0011	Medium	0.2577	0.0022
High	0.3187	0.0007	High	0.2582	0.0013

(a) Brier Results for Women Model

NHANES Estimate for Brier Score: 0.1402

Framingham Brier Score: 0.1160

(b) Brier Results for Men Model

NHANES Estimate for Brier Score: 0.1583

Framingham Brier Score: 0.1919

Table 2: 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

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. Particularly, 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 score surpassed that of the source population, yet notably diverged by gender. Specifically, in men, the Brier score was lower than that of the source population. This discrepancy 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. This suggests a potential inefficiency in the model’s transportability within the simulated target population, particularly for women. 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. Nonetheless, to enhance the model’s performance within these simulated populations, tailoring the model specifically to these contexts could be pivotal since a Brier score of approximately 0.3 implies a moderate level of inaccuracy or uncertainty in the model’s predictions. This score indicates that the model’s forecasted probabilities for specific events or outcomes exhibit notable discrepancies from the actual observed outcomes. 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 Steingrimsdottir, 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

```
1 library(riskCommunicator)
2 library(tidyverse)
3 library(tableone)
4 library(naniar)
5 library(dplyr)
6 library(MASS)
7 library(corrplot)
8 library(GGally)
9 library(ggpubr)
10 library(nhanesA)
11 library(ggcorrplot)
12 library(kableExtra)
```

```
1 data("framingham")
2 framingham_df <- framingham %>% dplyr::select(c(CVD, TIMECVD, SEX, TOTCHOL,
3       AGE,
4       SYSBP, DIABP, CURSMOKE, DIABETES, BPMEDS
5       ,
6       HDLC, BMI))
7 framingham_df <- na.omit(framingham_df)
8
9 # Get blood pressure based on whether or not on BPMEDS
10 framingham_df$SYSBP_UT <- ifelse(framingham_df$BPMEDS == 0,
11       framingham_df$SYSBP, 0)
12 framingham_df$SYSBP_T <- ifelse(framingham_df$BPMEDS == 1,
13       framingham_df$SYSBP, 0)
14
15 # Looking at risk within 15 years - remove censored data
16 framingham_df <- framingham_df %>%
17   filter(!(CVD == 0 & TIMECVD <= 365*15)) %>%
18   dplyr::select(-c(TIMECVD))
19
20 #Factor variables
21 framingham_df <- framingham_df %>%
22   mutate(SEX = as.factor(SEX),
23     CURSMOKE = as.factor(CURSMOKE),
24     DIABETES = as.factor(DIABETES),
25     BPMEDS = as.factor(BPMEDS)
26   )
27
28 # Filter to each sex
```

```

27 framingham_df_men <- framingham_df %>% filter(SEX == 1)
28 framingham_df_women <- framingham_df %>% filter(SEX == 2)
29
30 # Fit models with log transforms for all continuous variables
31 mod_men <- glm(as.factor(CVD)~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
32               log(SYSBP_T+1)+CURSMOKE+DIABETES,
33               data= framingham_df_men, family= "binomial")
34
35 mod_women <- glm(as.factor(CVD)~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT
36               +1)+
37               log(SYSBP_T+1)+CURSMOKE+DIABETES,
38               data= framingham_df_women, family= "binomial")

```

```

1   # The NHANES data here finds the same covariates among this national
2     survey data
3
4 # blood pressure, demographic, bmi, smoking, and hypertension info
5 bpx_2017 <- nhanes("BPX_J") %>%
6   dplyr::select(SEQN, BPXS1 ) %>%
7   rename(SYSBP = BPXS1)
8 demo_2017 <- nhanes("DEMO_J") %>%
9   dplyr::select(SEQN, RIAGENDR, RIDAGEYR) %>%
10  rename(SEX = RIAGENDR, AGE = RIDAGEYR)
11 bmx_2017 <- nhanes("BMX_J") %>%
12   dplyr::select(SEQN, BMXBMI) %>%
13   rename(BMI = BMXBMI)
14 smq_2017 <- nhanes("SMQ_J") %>%
15   mutate(CURSMOKE = case_when(SMQ040 %in% c(1,2) ~ 1,
16                               SMQ040 == 3 ~ 0,
17                               SMQ020 == 2 ~ 0)) %>%
18   dplyr::select(SEQN, CURSMOKE)
19 bpq_2017 <- nhanes("BPQ_J") %>%
20   mutate(BPMEDS = case_when(
21     BPQ020 == 2 ~ 0,
22     BPQ040A == 2 ~ 0,
23     BPQ050A == 1 ~ 1,
24     TRUE ~ NA )) %>%
25   dplyr::select(SEQN, BPMEDS)
26 tchol_2017 <- nhanes("TCHOL_J") %>%
27   dplyr::select(SEQN, LBXTC) %>%
28   rename(TOTCHOL = LBXTC)
29 hdl_2017 <- nhanes("HDL_J") %>%

```

```

30   dplyr::select(SEQN, LBDHDD) %>%
31   rename(HDLC = LBDHDD)
32   diq_2017 <- nhanes("DIQ_J") %>%
33   mutate(DIABETES = case_when(DIQ010 == 1 ~ 1,
34                               DIQ010 %in% c(2,3) ~ 0,
35                               TRUE ~ NA)) %>%
36   dplyr::select(SEQN, DIABETES)
37
38   # Join data from different tables
39   df_2017 <- bpx_2017 %>%
40   full_join(demo_2017, by = "SEQN") %>%
41   full_join(bmx_2017, by = "SEQN") %>%
42   full_join(hdl_2017, by = "SEQN") %>%
43   full_join(smq_2017, by = "SEQN") %>%
44   full_join(bpq_2017, by = "SEQN") %>%
45   full_join(tchol_2017, by = "SEQN") %>%
46   full_join(diq_2017, by = "SEQN")
47
48   #Factor variables
49   df_2017 <- df_2017 %>%
50   mutate(SEX = as.factor(SEX),
51          CURSMOKE = as.factor(CURSMOKE),
52          DIABETES = as.factor(DIABETES),
53          BPMEDS = as.factor(BPMEDS)
54   )
55
56   # Get blood pressure based on whether or not on BPMEDS
57   df_2017$SYSBP_UT <- ifelse(df_2017$BPMEDS == 0,
58                             df_2017$SYSBP, 0)
59   df_2017$SYSBP_T <- ifelse(df_2017$BPMEDS == 1,
60                             df_2017$SYSBP, 0)
61
62   #Eligibility Criteria: Age between 31-81
63   df_2017 <- df_2017 %>% filter(AGE >= 30 & AGE <=81 )

```

```

1   #MISSING DATA
2   vis_miss(df_2017 %>% dplyr::select(HDLC, TOTCHOL, AGE, BPMEDS, SYSBP_UT, SYSBP_T,
3   CURSMOKE, DIABETES))
4
5   apply(df_2017, 2, function(x) sum(is.na(x))/nrow(df_2017))
6   sum(complete.cases(df_2017))/nrow(df_2017)
7
8   #Complete case Analysis

```

```

8 df_2017 <- na.omit(df_2017)

1 #Calculate Brier score for two models in framingham dataset.
2 #Men Brierscore
3 pred_prob_men <- predict(mod_men, framingham_df_men, type = "response")
4 brier_men_fram <- sum((framingham_df_men$CVD - pred_prob_men)^2)/nrow(
    framingham_df_men)
5
6 #Women Brierscore
7 pred_prob_women <- predict(mod_women, framingham_df_women, type = "response")
8 brier_women_fram <- sum((framingham_df_women$CVD - pred_prob_women)^2)/nrow(
    framingham_df_women)
9
10 c(brier_men_fram, brier_women_fram)

1 #Merge datasets while select only variables in model
2 #The variable S is indicator for individual being on the framingham dataset
3 fram_df_men <- framingham_df_men %>%
4   dplyr::select(CVD, HDLC, TOTCHOL, AGE, SYSBP_UT, SYSBP_T, CURSMOKE, DIABETES)
5 y_men <- framingham_df_men %>% dplyr::select(CVD)
6 fram_df_men$S <- 1
7
8 fram_df_women <- framingham_df_women %>%
9   dplyr::select(CVD, HDLC, TOTCHOL, AGE, SYSBP_UT, SYSBP_T, CURSMOKE, DIABETES)
10 y_women <- framingham_df_women %>% dplyr::select(CVD)
11 fram_df_women$S <- 1
12
13 df_2017 <- df_2017 %>%
14   dplyr::select(HDLC, TOTCHOL, AGE, SYSBP_UT, SYSBP_T, CURSMOKE, DIABETES, SEX)
15 df_2017$S <- 0
16 df_2017$CVD <- NA
17
18 #re-order variables
19 df_2017 <- df_2017 %>%
20   dplyr::select(CVD, HDLC, TOTCHOL, AGE, SYSBP_UT, SYSBP_T, CURSMOKE, DIABETES, SEX, S)
21
22
23 nhanes_men <- df_2017 %>%
24   filter(SEX == "1") %>%
25   dplyr::select(!SEX)
26
27 nhanes_women <- df_2017 %>%
28   filter(SEX == "2") %>%

```

```

29   dplyr::select(!SEX)
30
31   #merge data
32   join_men <- rbind(fram_df_men, nhanes_men)
33   join_women <- rbind(fram_df_women, nhanes_women)
34
35   #factor S
36   join_men$S <- as.factor(join_men$S)
37   join_women$S <- as.factor(join_women$S)
38
39   #
40   #####
41
42   #####MEN
43   #FIT Inverse Odds
44   inv_odd_men <- glm(as.factor(S) ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT
45     +1)+
46       log(SYSBP_T+1)+CURSMOKE+DIABETES,
47       data = join_men, family = "binomial")
48   #predict
49   prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
50   #weights
51   inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(1)
52   #attach to dataset
53   join_men$pred_prob_weighted <- inv_prob_fram_men
54   join_men$pred_prob <- prob_fram_men
55
56   #calculate estimate of brier score
57   brier_men_nhanes_2 <- join_men %>%
58     filter(S == 1) %>%
59     summarise(brier_num = sum(pred_prob_weighted * (pred_prob - CVD)^2)) %>%
60     pull(brier_num) / join_men %>% filter(S==0) %>% count()
61
62   #another way to calculate (gives the same)
63   nhanes_men_brier <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
64     prob_fram_men[1:length(y_men$CVD)])^2)/nrow(nhanes_men)
65
66   #####WOMEN
67   #FIT Inverse Odds
68   #Do the same:
69   inv_odd_women <- glm(as.factor(S) ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_
70     UT+1)+
71       log(SYSBP_T+1)+CURSMOKE+DIABETES ,data = join_women,

```



```

                                family = "binomial")
67 prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
68 inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^-1)
69 join_women$pred_prob_weighted <- inv_prob_fram_women
70 join_women$pred_prob <- prob_fram_women
71
72
73
74 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)
75
76 brier_women_nhanes_2 <- join_women %>%
77   filter(S == 1) %>%
78   summarise(brier_num = sum(pred_prob_weighted * (pred_prob - CVD)^2)) %>%
79   pull(brier_num) / join_women %>% filter(S==0) %>% count()
80
81
82 #both estimates of brier scores
83 c(nhanes_men_brier,nhanes_women_brier)
84 c(brier_men_nhanes_2,brier_women_nhanes_2)

```

```

1 #MODEL USES: TOTCHOL,AGE,SYSBP,DIABP,HLDC SYSBP_T, SYSBP_UT, CURSMOKE and
  DIABETES
2 #SYSBP_UT and SYSBP_T are based on categorical variable BPMEDS
3
4 #Here we plot distributions of numeric variables of framinham dataset
5 #Continuous: TOTCHOL, AGE, DIABP, HDLC SYSSBP_T SYSBP_UT
6
7 ##HLDC
8 hdlc_women <- ggplot(fram_df_women, aes(x = log(HDLC)))+geom_density()+
9   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$HDLC)),
10     sd = sd(log(fram_df_women$HDLC))), color = "red")+
11   theme_minimal() +
12   ggtitle("Log(HDLC)")
13 hdlc_men <- ggplot(fram_df_men, aes(x = log(HDLC)))+geom_density()+
14   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$HDLC)),
15     sd = sd(log(fram_df_men$HDLC))), color = "red")+
16   theme_minimal() +
17   ggtitle("Log(HDLC)")
18
19 ##TOTCHOL
20 totchol_women <- ggplot(fram_df_women, aes(x = log(TOTCHOL)))+geom_density()+

```

```

20   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$TOTCHOL
    )), sd = sd(log(fram_df_women$TOTCHOL))), color = "red")+
21   theme_minimal() +
22   ggtitle("Log(TOTCHOL)")
23 totchol_men <- ggplot(fram_df_men, aes(x = log(TOTCHOL)))+geom_density()+
24   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$TOTCHOL))
    , sd = sd(log(fram_df_men$TOTCHOL))), color = "red")+
25   theme_minimal() +
26   ggtitle("Log(TOTCHOL)")
27
28 ##AGE
29 age_women <- ggplot(fram_df_women, aes(x = log(AGE)))+geom_density()+
30   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$AGE)),
    sd = sd(log(fram_df_women$AGE))), color = "red")+
31   theme_minimal() +
32   ggtitle("Log(AGE)")
33 age_men <- ggplot(fram_df_men, aes(x = log(AGE)))+geom_density()+
34   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_men$AGE)), sd
    = sd(log(fram_df_men$AGE))), color = "red")+
35   theme_minimal() +
36   ggtitle("Log(AGE)")
37
38
39
40 sys_women <- ggplot(fram_df_women, aes(x = log(AGE)))+geom_density()+
41   stat_function(fun = dnorm, args = list(mean = mean(log(fram_df_women$AGE)),
    sd = sd(log(fram_df_women$AGE))), color = "red")+
42   theme_minimal() +
43   ggtitle("Log(AGE)")
44
45 ##SYSBP
46 xx <- fram_df_women %>% dplyr::select(SYSBP_T) %>% filter(SYSBP_T != 0)
47 xx <- as.data.frame(xx)
48 sysbpT_women <- ggplot(xx, aes(log(SYSBP_T)))+geom_density()+
49   stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_T)), sd = sd
    (log(xx$SYSBP_T))), color = "red")+
50   theme_minimal() +
51   ggtitle("Log(SYSBP_T)")
52
53 xx <- fram_df_men %>% dplyr::select(SYSBP_T) %>% filter(SYSBP_T != 0)
54 xx <- as.data.frame(xx)
55 sysbpT_men <- ggplot(xx, aes(log(SYSBP_T)))+geom_density()+
56   stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_T)), sd = sd

```

```

    (log(xx$SYSBP_T))), color = "red")+
57 theme_minimal() +
58 ggtitle("Log(SYSBP_T)")
59
60 ##SYSBP_UT
61 xx <- fram_df_women %>% dplyr::select(SYSBP_UT) %>% filter(SYSBP_UT != 0)
62 xx <- as.data.frame(xx)
63 sysbpUT_women <- ggplot(xx,aes(log(SYSBP_UT)))+geom_density()+
64   stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_UT)), sd =
        sd(log(xx$SYSBP_UT))), color = "red")+
65   theme_minimal() +
66   ggtitle("Log(SYSBP_UT)")
67
68 xx <- fram_df_men %>% dplyr::select(SYSBP_UT) %>% filter(SYSBP_UT != 0)
69 xx <- as.data.frame(xx)
70 sysbpUT_men <- ggplot(xx,aes(log(SYSBP_UT)))+geom_density()+
71   stat_function(fun = dnorm, args = list(mean = mean(log(xx$SYSBP_UT)), sd =
        sd(log(xx$SYSBP_UT))), color = "red")+
72   theme_minimal() +
73   ggtitle("Log(SYSBP_UT)")
74
75 p1 <- ggarrange(hdlc_men,totchol_men,age_men,sysbpT_men,sysbpUT_men,nrow = 5)
    %>% annotate_figure(top = text_grob("Women"))
76 p2 <- ggarrange(hdlc_women,totchol_women,age_women,sysbpT_women,sysbpUT_women,
    nrow=5) %>% annotate_figure(top = text_grob("Men"))
77
78 ggarrange(p1,p2,ncol = 2) %>% annotate_figure(top = text_grob("Continuous
    Densities with Normal Approximations", size = 14, face = "bold"))

```

```

1 #
    #####
2
3 #FUNCTION TO GENERATE WHILE VARYING CORRELATIONS WOMEN
4
5 sim_data_function_men <- function(data,mu,cov){
6
7 framingham_df_men_log <- data %>%
8   mutate(log_HDLC = log(HDLC),
9     log_TOTCHOL = log(TOTCHOL),
10     log_AGE = log(AGE),
11     log_SYSBP = log(SYSBP)) %>%
12   dplyr::select(log_HDLC,log_TOTCHOL,log_AGE,log_SYSBP,BPMEDS,CURSMOKE,

```

```

    DIABETES)
13
14
15 sim_data <- mvrnorm(n = 1000, mu = mu, Sigma = cov)
16 sim_data <- as.data.frame(sim_data)
17
18 #CHANGE BPMEDS TO CATEGORICAL
19 pp <- sum(as.numeric(data$BPMEDS)-1)/nrow(data)
20 sim_data$BPMEDS <- ifelse(sim_data$BPMEDS > quantile(sim_data$BPMEDS, 1 - pp),
    1, 0)
21
22 #CHANGE DIABETES TO CATEGORICAL
23 pp <- sum(as.numeric(data$DIABETES)-1)/nrow(data)
24 sim_data$DIABETES <- ifelse(sim_data$DIABETES > quantile(sim_data$DIABETES, 1
    - pp), 1, 0)
25
26 #CHANGE CURSMOKE TO CATEGORICAL
27 pp <- sum(as.numeric(data$CURSMOKE)-1)/nrow(data)
28 sim_data$CURSMOKE <- ifelse(sim_data$CURSMOKE > quantile(sim_data$CURSMOKE, 1
    - pp), 1, 0)
29
30
31 return(sim_data)
32 }
33
34
35 #FUNCTION TO GENERATE WHILE VARYING CORRELATIONS WOMEN
36
37 sim_data_function_women <- function(data, mu, cov){
38
39   framingham_df_women_log <- data %>%
40   mutate(log_HDLC = log(HDLC),
41          log_TOTCHOL = log(TOTCHOL),
42          log_AGE = log(AGE),
43          log_SYSBP = log(SYSBP)) %>%
44   dplyr::select(log_HDLC, log_TOTCHOL, log_AGE, log_SYSBP, BPMEDS, CURSMOKE,
    DIABETES)
45
46 sim_data <- mvrnorm(n = 1000, mu = mu, Sigma = cov)
47 sim_data <- as.data.frame(sim_data)
48
49 #CHANGE BPMEDS TO CATEGORICAL
50 pp <- sum(as.numeric(data$BPMEDS)-1)/nrow(data)

```

```

51 sim_data$BPMEDS <- ifelse(sim_data$BPMEDS > quantile(sim_data$BPMEDS, 1 - pp),
    1, 0)
52
53 #CHANGE DIABETES TO CATEGORICAL
54 pp <- sum(as.numeric(data$DIABETES)-1)/nrow(data)
55 sim_data$DIABETES <- ifelse(sim_data$DIABETES > quantile(sim_data$DIABETES, 1
    - pp), 1, 0)
56
57 #CHANGE CURSMOKE TO CATEGORICAL
58 pp <- sum(as.numeric(data$CURSMOKE)-1)/nrow(data)
59 sim_data$CURSMOKE <- ifelse(sim_data$CURSMOKE > quantile(sim_data$CURSMOKE, 1
    - pp), 1, 0)
60
61 return(sim_data)
62 }

```

```

1 ##HIGLY CORRELATED DATASET
2 #https://stats.stackexchange.com/questions/124538/how-to-generate-a-large-full
    -rank-random-correlation-matrix-with-some-strong-cor
3 # Generate W matrix with random values from a standard normal distribution
4
5 v <- 7          # 12 variables
6 f <- 5          # Subset-correlation
7 vg <- v / f     # Variables per subset
8 a <- 5
9 d <- 100
10 # Constructing a factor matrix 'L' with higher magnitude positive and negative
    relationships
11 set.seed(3)
12 L <- matrix(c(
13   runif(vg*f, -a, a), runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d,
14   runif(vg*f, -a, a)/d, runif(vg*f, -a, a), runif(vg*f, -a, a)/d,
15   runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d, runif(vg*f, -a, a)
16 ), nrow = v, ncol = v)
17
18 names <- c("log_HDL", "log_TOTCHOL", "log_AGE", "log_SYSBP", "BPMEDS", "CURSMOKE",
    "DIABETES")
19 colnames(L) <- names
20 rownames(L) <- names
21
22 # Make covariance and correlation matrix
23 cov_matrix <- L %*% t(L)
24 colnames(cov_matrix) <- names

```

```

25 rownames(cov_matrix) <- names # L multiplied with its transpose
26 cor_high<- cov2cor(cov_matrix)

```

```

1 ##Medium CORRELATED DATASET
2 #https://stats.stackexchange.com/questions/124538/how-to-generate-a-large-full
   -rank-random-correlation-matrix-with-some-strong-cor
3 # Generate W matrix with random values from a standard normal distribution
4
5 v <- 7           # 12 variables
6 f <- 1           # Subset-correlation based on 5 common factors
7 vg <- v / f      # Variables per subset
8 a <- .1
9 d <- 5
10
11 # Constructing a factor matrix 'L' with higher magnitude positive and negative
   relationships
12 set.seed(1)
13 L <- matrix(c(
14   runif(vg*f, -a, a), runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d,
15   runif(vg*f, -a, a)/d, runif(vg*f, -a, a), runif(vg*f, -a, a)/d,
16   runif(vg*f, -a, a)/d, runif(vg*f, -a, a)/d, runif(vg*f, -a, a)
17 ), nrow = v, ncol = v)
18
19
20 colnames(L) <- names
21 rownames(L) <- names
22
23 # Make covariance and correlation matrix
24 cov_matrix <- L %%% t(L)
25 colnames(cov_matrix) <- names
26 rownames(cov_matrix) <- names # L multiplied with its transpose
27 cor_med <- cov2cor(cov_matrix)
28
29 ggcorrplot(cor_med)

```

```

1 #Statistic from dataset
2
3 #MEN
4 framingham_df_men_log <- framingham_df_men %>%
5   mutate(log_HDL = log(HDL),
6           log_TOTCHOL = log(TOTCHOL),
7           log_AGE = log(AGE),
8           log_SYSBP = log(SYSBP)) %>%

```

```

9   dplyr::select(log_HDL, log_TOTCHOL, log_AGE, log_SYSBP, BPMEDS, CURSMOKE,
10                  DIABETES)
11 mu_men <- framingham_df_men_log %>%
12   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
13          DIABETES = as.numeric(DIABETES)-1,
14          BPMEDS = as.numeric(BPMEDS)-1) %>%
15   summarize_if(is.numeric, mean)
16
17 sd_men <- framingham_df_men_log %>%
18   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
19          DIABETES = as.numeric(DIABETES)-1,
20          BPMEDS = as.numeric(BPMEDS)-1) %>%
21   summarize_if(is.numeric, sd)
22
23 #WOMEN
24 framingham_df_women_log <- framingham_df_women %>%
25   mutate(log_HDL = log(HDL),
26          log_TOTCHOL = log(TOTCHOL),
27          log_AGE = log(AGE),
28          log_SYSBP = log(SYSBP)) %>%
29   dplyr::select(log_HDL, log_TOTCHOL, log_AGE, log_SYSBP, BPMEDS, CURSMOKE,
30                  DIABETES)
31 mu_women <- framingham_df_women_log %>%
32   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
33          DIABETES = as.numeric(DIABETES)-1,
34          BPMEDS = as.numeric(BPMEDS)-1) %>%
35   summarize_if(is.numeric, mean)
36
37 sd_women <- framingham_df_women_log %>%
38   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
39          DIABETES = as.numeric(DIABETES)-1,
40          BPMEDS = as.numeric(BPMEDS)-1) %>%
41   summarize_if(is.numeric, sd)
42
43
44
45 #NO CORRELATION
46 cor_no <- diag(1,7)
47 colnames(cor_no) <- names
48 rownames(cor_no) <- names
49

```

```

50 cov_no <- diag(sd_men) %*% cor_no %*% diag(sd_men)
51 colnames(cov_no) <- names
52 rownames(cov_no) <- names
53
54 cov_no_women <- diag(sd_women) %*% cor_no %*% diag(sd_women)
55 colnames(cov_no_women) <- names
56 rownames(cov_no_women) <- names

1 p.cor.high <- ggcorrplot(cor_high) + ggtitle("High Correlation")
2 p.cor.med <- ggcorrplot(cor_med) + ggtitle("Medium Correlation")
3 p.cor.no <- ggcorrplot(cor_no) + ggtitle("No Correlation")
4
5 ggarrange(p.cor.no,p.cor.med,p.cor.high,common.legend = T)
6
7 ggarrange(ggarrange(p.cor.no,p.cor.med,ncol = 2, legend = "none"),p.cor.high,
8           nrow = 2, legend = "bottom")
9 #####
10 #SIMILAR CORRELATION WITH FRAMINGHAM DATASET
11 cov_men <- framingham_df_men_log %>%
12   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
13          DIABETES = as.numeric(DIABETES)-1,
14          BPMEDS = as.numeric(BPMEDS)-1) %>%
15   cov()
16
17
18 sim_data_similar_men <- sim_data_function_men(framingham_df_men, as.numeric(mu
19   _men), cov_men)
20
21 # Get blood pressure based on whether or not on BPMEDS on simulated data
22 sim_data_similar_men$log_SYSBP_UT <- ifelse(sim_data_similar_men$BPMEDS == 0,
23   sim_data_similar_men$log_SYSBP, 0)
24 sim_data_similar_men$log_SYSBP_T <- ifelse(sim_data_similar_men$BPMEDS == 1,
25   sim_data_similar_men$log_SYSBP, 0)
26
27 sim_data_similar_men_2 <- sim_data_similar_men %>%
28   dplyr::select(log_HDL, log_TOTCHOL, log_AGE, log_SYSBP_UT, log_SYSBP_T,
29     CURSMOKE, DIABETES)
30
31 p1 <- sim_data_similar_men %>%
32   dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
33   cor() %>%
34   ggcorrplot()+ ggtitle("MEN Framingham Simulated Correlation")

```



```

33
34 p <- framingham_df_men_log %>%
35   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
36          DIABETES = as.numeric(DIABETES)-1,
37          BPMEDS = as.numeric(BPMEDS)-1) %>%
38   cor() %>%
39   ggcorrplot()+ggtitle("Framingham Correlation") #same summary statistics
40
41
42 ##SIMULATION
43 brier_similar_men <- numeric(0)
44 for(i in 1:5000){
45   sim_data_similar_men <- sim_data_function_men(framingham_df_men, as.numeric(mu
46     _men), cov_men)
47   # Get blood pressure based on whether or not on BPMEDS on simulated data
48   sim_data_similar_men$log_SYSBP_UT <- ifelse(sim_data_similar_men$BPMEDS == 0,
49     sim_data_similar_men$log_SYSBP, 0)
50   sim_data_similar_men$log_SYSBP_T <- ifelse(sim_data_similar_men$BPMEDS == 1,
51     sim_data_similar_men$log_SYSBP, 0)
52
53   ##ESTIMATE BRIER SCORE
54   ####MEN
55   framingham_df_men_log$S <- 1
56   #FIT Inverse Odds
57   framingham_df_men_log$log_SYSBP_UT <- ifelse(framingham_df_men_log$BPMEDS ==
58     0,
59     framingham_df_men_log$log_SYSBP, 0)
60   framingham_df_men_log$log_SYSBP_T <- ifelse(framingham_df_men_log$BPMEDS == 1,
61     framingham_df_men_log$log_SYSBP, 0)
62
63   sim_data_similar_men$S <- 0
64   join_men <- rbind(framingham_df_men_log, sim_data_similar_men)
65   inv_odd_men <- glm(as.factor(S) ~ log_HDL + log_TOTCHOL + log_AGE + log_SYSBP
66     _UT
67     +log_SYSBP_T + CURSMOKE+DIABETES,
68     data = join_men, family = "binomial")
69   prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
70   inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(1)
71
72   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)
73 print(i)
74 }
75 brier_similar_men_avg <- mean(brier_similar_men)
76 brier_similar_men_sd <- sd(brier_similar_men)
77 pbrier_similar_men <- ggplot(as.data.frame(brier_similar_men), aes(x = brier_
    similar_men)) +
78   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
    width
79   labs(
80     title = "Similar Correlation",
81     x = "Brier Score",
82     y = "Frequency") +
83   theme_minimal()

```

```

1 ##### GENERATE DATA
2
3 ## HIGH CORRELATION
4 cov_men <- diag(sd_men) %*% cor_high %*% diag(sd_men)
5 colnames(cov_men) <- names
6 rownames(cov_men) <- names
7
8 sim_data_high_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
    men), cov_men)
9
10
11 # Get blood pressure based on whether or not on BPMEDS on simulated data
12 sim_data_high_men$log_SYSBP_UT <- ifelse(sim_data_high_men$BPMEDS == 0,
13     sim_data_high_men$log_SYSBP, 0)
14 sim_data_high_men$log_SYSBP_T <- ifelse(sim_data_high_men$BPMEDS == 1,
15     sim_data_high_men$log_SYSBP, 0)
16
17 p2 <- sim_data_high_men %>%
18   dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
19   cor() %>%
20   ggcorrplot() + ggtitle("High Correlation")
21
22 ## SIMULATION
23 brier_high_men <- numeric(0)
24 for(i in 1:5000){
25   sim_data_high_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
        men), cov_men)
26

```

```

27
28
29 # Get blood pressure based on whether or not on BPMEDS on simulated data
30 sim_data_high_men$log_SYSBP_UT <- ifelse(sim_data_high_men$BPMEDS == 0,
31                                           sim_data_high_men$log_SYSBP, 0)
32 sim_data_high_men$log_SYSBP_T <- ifelse(sim_data_high_men$BPMEDS == 1,
33                                           sim_data_high_men$log_SYSBP, 0)
34
35
36 ##ESTIMATE BRIER SCORE
37 #####MEN
38 #FIT Inverse Odds
39 framingham_df_men_log$S <- 1
40 sim_data_high_men$S <- 0
41
42 join_men <- rbind(framingham_df_men_log, sim_data_high_men)
43 inv_odd_men <- glm(as.factor(S) ~ log_HDL + log_TOTCHOL + log_AGE + log_SYSBP
44                   _UT
45                   +log_SYSBP_T + CURSMOKE+DIABETES,
46                   data = join_men, family = "binomial")
47 prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
48 inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^(-1)
49
50 brier_high_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
51   prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_high_men)
52 print(i)
53 }
54
55 brier_high_men_avg <- mean(brier_high_men)
56 brier_high_men_sd <- sd(brier_high_men)
57 pbrier_high_men <- ggplot(as.data.frame(brier_high_men),aes(x = brier_high_men
58   ))+
59   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
60   width
61   labs(
62     title = "High Correlation",
63     x = "Brier Score",
64     y = "Frequency")+
65   theme_minimal()
66
67 #####
68 #Medium

```

```

66 ##GENERATE DATA
67
68 cov_men <- diag(sd_men) %*% cor_med %*% diag(sd_men)
69 colnames(cov_men) <- names
70 rownames(cov_men) <- names
71
72 sim_data_med_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_men
    ), cov_men)
73
74
75 # Get blood pressure based on whether or not on BPMEDS on simulated data
76 sim_data_med_men$log_SYSBP_UT <- ifelse(sim_data_med_men$BPMEDS == 0,
77     sim_data_med_men$log_SYSBP, 0)
78 sim_data_med_men$log_SYSBP_T <- ifelse(sim_data_med_men$BPMEDS == 1,
79     sim_data_med_men$log_SYSBP, 0)
80
81 p3 <- sim_data_med_men %>%
82     dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
83     cor() %>%
84     ggcorrplot()+ ggtitle("Medium Correlation")
85
86 ##SIMULATION
87 brier_med_men <- numeric(0)
88 for(i in 1:5000){
89     sim_data_med_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_men
90         ), cov_men)
91
92 # Get blood pressure based on whether or not on BPMEDS on simulated data
93 sim_data_med_men$log_SYSBP_UT <- ifelse(sim_data_med_men$BPMEDS == 0,
94     sim_data_med_men$log_SYSBP, 0)
95 sim_data_med_men$log_SYSBP_T <- ifelse(sim_data_med_men$BPMEDS == 1,
96     sim_data_med_men$log_SYSBP, 0)
97 ##ESTIMATE BRIER SCORE
98 #####MEN
99 #FIT Inverse Odds
100 sim_data_med_men$S <- 0
101
102
103
104 join_men <- rbind(framingham_df_men_log, sim_data_med_men)
105 inv_odd_men <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_SYSBP
    _UT

```

```

106         +log_SYSBP_T + CURSMOKE+ DIABETES,
107         data = join_men, family = "binomial")
108 prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
109 inv_prob_fram_men <- (prob_fram_men/(1-prob_fram_men))^-1
110
111
112 brier_med_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)]*(y_men$CVD -
113     prob_fram_men[1:length(y_men$CVD)])^2)/nrow(sim_data_med_men)
114 print(i)
115 }
116
117 brier_med_men_avg <- mean(brier_med_men)
118 brier_med_men_sd <- sd(brier_med_men)
119 pbrier_med_men <- ggplot(as.data.frame(brier_med_men),aes(x = brier_med_men))+
120     geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
121     width
122     labs(
123         title = "Medium Correlation",
124         x = "Brier Score",
125         y = "Frequency")+
126     theme_minimal()
127
128 ##NO CORR
129 sim_data_nocor_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
130     men), cov_no)
131
132 # Get blood pressure based on whether or not on BPMEDS on simulated data
133 sim_data_nocor_men$log_SYSBP_UT <- ifelse(sim_data_nocor_men$BPMEDS == 0,
134     sim_data_nocor_men$log_SYSBP, 0)
135 sim_data_nocor_men$log_SYSBP_T <- ifelse(sim_data_nocor_men$BPMEDS == 1,
136     sim_data_nocor_men$log_SYSBP, 0)
137
138 p4 <- sim_data_nocor_men %>%
139     dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
140     cor() %>%
141     ggcorrplot()+ ggtitle("No Correlation (Men)")
142
143 #SIMULATION
144 brier_nocor_men <- numeric(0)
145 for(i in 1:5000){
146     sim_data_nocor_men <- sim_data_function_men(framingham_df_men, as.numeric(mu_
147         men), cov_no)

```

```

145
146
147 # Get blood pressure based on whether or not on BPMEDS on simulated data
148 sim_data_nocor_men$log_SYSBP_UT <- ifelse(sim_data_nocor_men$BPMEDS == 0,
149                                           sim_data_nocor_men$log_SYSBP, 0)
150 sim_data_nocor_men$log_SYSBP_T <- ifelse(sim_data_nocor_men$BPMEDS == 1,
151                                           sim_data_nocor_men$log_SYSBP, 0)
152 ##ESTIMATE BRIER SCORE
153 ####MEN
154 #FIT Inverse Odds
155 sim_data_nocor_men$S <- 0
156
157 join_men <- rbind(framingham_df_men_log, sim_data_nocor_men)
158 inv_odd_men <- glm(as.factor(S) ~ log_HDL + log_TOTCHOL + log_AGE + log_SYSBP
159                   _UT
160                   + log_SYSBP_T + CURSMOKE + DIABETES,
161                   data = join_men, family = "binomial")
162 prob_fram_men <- predict(inv_odd_men, join_men, type = "response")
163 inv_prob_fram_men <- (prob_fram_men / (1 - prob_fram_men)) ^ (-1)
164
165 brier_nocor_men[i] <- sum(inv_prob_fram_men[1:length(y_men$CVD)] * (y_men$CVD -
166   prob_fram_men[1:length(y_men$CVD)])^2) / nrow(sim_data_nocor_men)
167 print(i)
168 }
169
170 brier_nocor_men_avg <- mean(brier_nocor_men)
171 brier_nocor_men_sd <- sd(brier_nocor_men)
172 pbrier_nocor_men <- ggplot(as.data.frame(brier_nocor_men), aes(x = brier_nocor_
173   men)) +
174   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
175   width
176   labs(
177     title = "Independent Correlation",
178     x = "Brier Score",
179     y = "Frequency") +
180   theme_minimal()

```

```

1 #
  #####
  WOMEN
  #####

```

```

2
3
4
5 #####
6 #SIMILAR CORRELATION WITH FRAMINGHAM DATASET
7 cov_women <- framingham_df_women_log %>%
8   dplyr::select(names) %>%
9   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
10          DIABETES = as.numeric(DIABETES)-1,
11          BPMEDS = as.numeric(BPMEDS)-1) %>%
12   cov()
13
14
15 sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
16   numeric(mu_women), cov_women)
17
18 # Get blood pressure based on whether or not on BPMEDS on simulated data
19 sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS ==
20   0,
21   sim_data_similar_women$log_SYSBP,
22   0)
23
24 sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS ==
25   1,
26   sim_data_similar_women$log_SYSBP,
27   0)
28
29 p1 <- sim_data_similar_women %>%
30   dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
31   cor() %>%
32   ggcorrplot()+ ggtitle("women Framingham Simulated Correlation")
33
34 p <- framingham_df_women_log %>%
35   dplyr::select(-S)%>%
36   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
37          DIABETES = as.numeric(DIABETES)-1,
38          BPMEDS = as.numeric(BPMEDS)-1) %>%
39   cor() %>%
40   ggcorrplot()+ggtitle("Framingham Correlation") #same summary statistics
41
42 ##SIMULATION
43 brier_similar_women <- numeric(0)

```

```

40 for(i in 1:5000){
41   sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
      numeric(mu_women), cov_women)
42
43   # Get blood pressure based on whether or not on BPMEDS on simulated data
44   sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS
      == 0,
45
      sim_data_similar_women$log_SYSBP
      , 0)
46   sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS
      == 1,
47
      sim_data_similar_women$log_SYSBP,
      0)
48
49
50   ##ESTIMATE BRIER SCORE
51   ####women
52   #FIT Inverse Odds
53   framingham_df_women_log$log_SYSBP_UT <- ifelse(framingham_df_women_log$
      BPMEDS == 0,
54
      framingham_df_women_log$log_
      SYSBP, 0)
55   framingham_df_women_log$log_SYSBP_T <- ifelse(framingham_df_women_log$BPMEDS
      == 1,
56
      framingham_df_women_log$log_
      SYSBP, 0)
57   framingham_df_women_log$S <- 1
58   sim_data_similar_women$S <- 0
59
60   join_women <- rbind(framingham_df_women_log, sim_data_similar_women)
61
62   inv_odd_women <- glm(as.factor(S) ~ log_HDLc + log_TOTCHOL + log_AGE +
      log_SYSBP_UT+log_SYSBP_T + CURSMOKE+DIABETES,
63
      data = join_women, family = "binomial")
64
65   prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
66   inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^-1
67
68
69   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)
70   print(i)
71

```



```

72 }
73 brier_similar_women_avg <- mean(brier_similar_women)
74 brier_similar_women_sd <- sd(brier_similar_women)
75 pbrier_similar_women <- ggplot(as.data.frame(brier_similar_women), aes(x =
    brier_similar_women)) +
76   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
    width
77   labs(
78     title = "Similar Correlation",
79     x = "Brier Score",
80     y = "Frequency") +
81   theme_minimal()
82
83
84
85
86 #####
87 #SIMILAR CORRELATION WITH FRAMINGHAM DATASET
88 cov_women <- framingham_df_women_log %>%
89   dplyr::select(names) %>%
90   mutate(CURSMOKE = as.numeric(CURSMOKE)-1,
91          DIABETES = as.numeric(DIABETES)-1,
92          BPMEDS = as.numeric(BPMEDS)-1) %>%
93   cov()
94
95
96 sim_data_similar_women <- sim_data_function_women(framingham_df_women, as.
    numeric(mu_women), cov_women)
97
98 # Get blood pressure based on whether or not on BPMEDS on simulated data
99 sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS ==
    0,
100                                             sim_data_similar_women$log_SYSBP,
    0)
101 sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS ==
    1,
102                                             sim_data_similar_women$log_SYSBP,
    0)
103
104 p1 <- sim_data_similar_women %>%
105   dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
106   cor() %>%
107   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.
122     numeric(mu_women), cov_women)
123
124   # Get blood pressure based on whether or not on BPMEDS on simulated data
125   sim_data_similar_women$log_SYSBP_UT <- ifelse(sim_data_similar_women$BPMEDS
126     == 0,
127     sim_data_similar_women$log_SYSBP
128     , 0)
129   sim_data_similar_women$log_SYSBP_T <- ifelse(sim_data_similar_women$BPMEDS
130     == 1,
131     sim_data_similar_women$log_SYSBP,
132     0)
133
134   ##ESTIMATE BRIER SCORE
135   ####women
136   #FIT Inverse Odds
137   framingham_df_women_log$log_SYSBP_UT <- ifelse(framingham_df_women_log$
138     BPMEDS == 0,
139     framingham_df_women_log$log_
140     SYSBP, 0)
141   framingham_df_women_log$log_SYSBP_T <- ifelse(framingham_df_women_log$BPMEDS
142     == 1,
143     framingham_df_women_log$log_
144     SYSBP, 0)
145
146   framingham_df_women_log$S <- 1
147   sim_data_similar_women$S <- 0
148
149   join_women <- rbind(framingham_df_women_log, sim_data_similar_women)
150
151   inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_

```

```

SYSBP_UT
142         +log_SYSBP_T + CURSMOKE+DIABETES,
143         data = join_women, family = "binomial")
144 prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
145 inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^(1)
146
147
148 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)
149 print(i)
150 }
151 brier_similar_women_avg <- mean(brier_similar_women)
152 brier_similar_women_sd <- sd(brier_similar_women)
153 pbrier_similar_women <- ggplot(as.data.frame(brier_similar_women),aes(x =
    brier_similar_women))+
154   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
    width
155   labs(
156     title = "Similar Correlation",
157     x = "Brier Score",
158     y = "Frequency")+
159   theme_minimal()

```

```

1 ##HIGH CORRELATION
2 cov_women <- diag(sd_women) %%% cor_high %%% diag(sd_women)
3 colnames(cov_women) <- names
4 rownames(cov_women) <- names
5
6 sim_data_high_women <- sim_data_function_women(framingham_df_women, as.numeric
    (mu_women), cov_women)
7
8
9 # Get blood pressure based on whether or not on BPMEDS on simulated data
10 sim_data_high_women$log_SYSBP_UT <- ifelse(sim_data_high_women$BPMEDS == 0,
11     sim_data_high_women$log_SYSBP, 0)
12 sim_data_high_women$log_SYSBP_T <- ifelse(sim_data_high_women$BPMEDS == 1,
13     sim_data_high_women$log_SYSBP, 0)
14
15 p2 <- sim_data_high_women %>%
16   dplyr::select(-log_SYSBP_UT,-log_SYSBP_T) %>%
17   cor() %>%
18   ggcorrplot()+ ggtitle("High Correlation")

```

```

19
20 ##SIMULATION
21 brier_high_women <- numeric(0)
22 for(i in 1:5000){
23   sim_data_high_women <- sim_data_function_women(framingham_df_women, as.
24     numeric(mu_women), cov_women)
25
26   # Get blood pressure based on whether or not on BPMEDS on simulated data
27   sim_data_high_women$log_SYSBP_UT <- ifelse(sim_data_high_women$BPMEDS == 0,
28     sim_data_high_women$log_SYSBP, 0)
29   sim_data_high_women$log_SYSBP_T <- ifelse(sim_data_high_women$BPMEDS == 1,
30     sim_data_high_women$log_SYSBP, 0)
31   ##ESTIMATE BRIER SCORE
32   ####women
33   #FIT Inverse Odds
34   sim_data_high_women$S <- 0
35
36   join_women <- rbind(framingham_df_women_log, sim_data_high_women)
37   inv_odd_women <- glm(as.factor(S) ~ log_HDL + log_TOTCHOL + log_AGE + log_
38     SYSBP_UT
39     +log_SYSBP_T + CURSMOKE+ DIABETES,
40     data = join_women, family = "binomial")
41   prob_fram_women <- predict(inv_odd_women, join_women, type = "response")
42   inv_prob_fram_women <- (prob_fram_women/(1-prob_fram_women))^-1
43
44   brier_high_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
45     women$CVD - prob_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
46     high_women)
47   print(i)
48 }
49
50 brier_high_women_avg <- mean(brier_high_women)
51 brier_high_women_sd <- sd(brier_high_women)
52 pbrier_high_women <- ggplot(as.data.frame(brier_high_women),aes(x = brier_high
53   _women))+
54   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
55   width
56   labs(
57     title = "High Correlation",
58     x = "Brier Score",
59     y = "Frequency")+

```

```

56   theme_minimal()
57
58   #####
59   #Medium
60   ##GENERATE DATA
61
62   cov_women <- diag(sd_women) %**% cor_med %**% diag(sd_women)
63   colnames(cov_women) <- names
64   rownames(cov_women) <- names
65
66   sim_data_med_women <- sim_data_function_women(framingham_df_women, as.numeric(
        mu_women), cov_women)
67
68
69   # Get blood pressure based on whether or not on BPMEDS on simulated data
70   sim_data_med_women$log_SYSBP_UT <- ifelse(sim_data_med_women$BPMEDS == 0,
71                                             sim_data_med_women$log_SYSBP, 0)
72   sim_data_med_women$log_SYSBP_T <- ifelse(sim_data_med_women$BPMEDS == 1,
73                                             sim_data_med_women$log_SYSBP, 0)
74
75   p3 <- sim_data_med_women %>%
76     dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
77     cor() %>%
78     ggcorrplot()+ ggtitle("Medium Correlation")
79
80   #SIMULATION
81   brier_med_women <- numeric(0)
82   for(i in 1:5000){
83     sim_data_med_women <- sim_data_function_women(framingham_df_women, as.
        numeric(mu_women), cov_women)
84
85
86     # Get blood pressure based on whether or not on BPMEDS on simulated data
87     sim_data_med_women$log_SYSBP_UT <- ifelse(sim_data_med_women$BPMEDS == 0,
88                                             sim_data_med_women$log_SYSBP, 0)
89     sim_data_med_women$log_SYSBP_T <- ifelse(sim_data_med_women$BPMEDS == 1,
90                                             sim_data_med_women$log_SYSBP, 0)
91
92     ##ESTIMATE BRIER SCORE
93     ####women
94     #FIT Inverse Odds
95     sim_data_med_women$S <- 0
96
97     join_women <- rbind(framingham_df_women_log, sim_data_med_women)

```

```

97   inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
      SYSBP_UT
98                               +log_SYSBP_T + CURSMOKE+ DIABETES,
99                               data = join_women, family = "binomial")
100   probb_fram_women <- predict(inv_odd_women, join_women, type = "response")
101   inv_prob_fram_women <- (probb_fram_women/(1-probb_fram_women))^-1)
102
103
104   brier_med_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
      women$CVD - probb_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_med
      _women)
105   print(i)
106 }
107
108 brier_med_women_avg <- mean(brier_med_women)
109 brier_med_women_sd <- sd(brier_med_women)
110 pbrier_med_women <- ggplot(as.data.frame(brier_med_women),aes(x = brier_med_
      women))+
111   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
      width
112   labs(
113     title = "Medium Correlation",
114     x = "Brier Score",
115     y = "Frequency")+
116   theme_minimal()
117
118
119 #####
120 #####
121 ##NO CORR
122 sim_data_nocor_women <- sim_data_function_women(framingham_df_women, as.
      numeric(mu_women), cov_no)
123
124
125 # Get blood pressure based on whether or not on BPMEDS on simulated data
126 sim_data_nocor_women$log_SYSBP_UT <- ifelse(sim_data_nocor_women$BPMEDS == 0,
      sim_data_nocor_women$log_SYSBP, 0)
127
128 sim_data_nocor_women$log_SYSBP_T <- ifelse(sim_data_nocor_women$BPMEDS == 1,
      sim_data_nocor_women$log_SYSBP, 0)
129
130
131 p4 <- sim_data_nocor_women %>%
132   dplyr::select(-log_SYSBP_UT, -log_SYSBP_T) %>%
133   cor() %>%

```

```

134 ggcorrplot()+ ggtitle("No Correlation (women)")
135
136 #SIMULATION
137 brier_nocor_women <- numeric(0)
138 for(i in 1:5000){
139   sim_data_nocor_women <- sim_data_function_women(framingham_df_women, as.
140     numeric(mu_women), cov_no)
141
142   # Get blood pressure based on whether or not on BPMEDS on simulated data
143   sim_data_nocor_women$log_SYSBP_UT <- ifelse(sim_data_nocor_women$BPMEDS ==
144     0,
145     sim_data_nocor_women$log_SYSBP, 0)
146   sim_data_nocor_women$log_SYSBP_T <- ifelse(sim_data_nocor_women$BPMEDS == 1,
147     sim_data_nocor_women$log_SYSBP, 0)
148
149   ##ESTIMATE BRIER SCORE
150   ###women
151   #FIT Inverse Odds
152   sim_data_nocor_women$S <- 0
153
154   join_women <- rbind(framingham_df_women_log, sim_data_nocor_women)
155   inv_odd_women <- glm(as.factor(S) ~ log_HDLC + log_TOTCHOL + log_AGE + log_
156     SYSBP_UT
157     +log_SYSBP_T + CURSMOKE+ DIABETES,
158     data = join_women, family = "binomial")
159   probb_fram_women <- predict(inv_odd_women, join_women, type = "response")
160   inv_prob_fram_women <- (probb_fram_women/(1-probb_fram_women))^-1
161
162   brier_nocor_women[i] <- sum(inv_prob_fram_women[1:length(y_women$CVD)]*(y_
163     women$CVD - probb_fram_women[1:length(y_women$CVD)])^2)/nrow(sim_data_
164     nocor_women)
165   print(i)
166 }
167
168 brier_nocor_women_avg <- mean(brier_nocor_women)
169 brier_nocor_women_sd <- sd(brier_nocor_women)
170 pbrier_nocor_women <- ggplot(as.data.frame(brier_nocor_women), aes(x = brier_
171   nocor_women))+
172   geom_histogram(color = "black", fill = "grey") + # Adjust color and bin
173     width
174   labs(
175     title = "Independent Correlation",

```

```

170   x = "Brier Score",
171   y = "Frequency")+
172   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))

```



```

5 res_men <- round(res_men,4)
6 colnames(res_men) <- c("Brier", "SD Brier")
7 rownames(res_men) <- c("Uncorrelated", "Similar", "Medium", "High")
8 tab_men <- res_men %>%
9   kable(booktabs = TRUE, caption = " Brier Results for Men Model") %>%
10   kableExtra::kable_styling(font_size = 8, latex_options = "HOLD_position")
11   %>% add_footnote(paste("NHANES Brier score is", round(nhanes_men_
12     brier,4)))
13
14 ggarrange(pbrier_nocor_women,pbrier_similar_women,pbrier_med_women,pbrier_high
15   _women,ncol = 2,nrow = 2) %>% annotate_figure(top = text_grob("Women
16   Distribution of Estimate for Brier Score on Matrix", size = 14, face = "
17   bold"))
18
19 res_women <- cbind(c(brier_nocor_women_avg, brier_similar_women_avg, brier_med
20   _women_avg, brier_high_women_avg),c(brier_nocor_women_sd, brier_similar_
21   women_sd, brier_med_women_sd, brier_high_women_sd))
22 res_women <- round(res_women,4)
23 colnames(res_women) <- c("Brier", "SD Brier")
24 rownames(res_women) <- c("Uncorrelated", "Similar", "Medium", "High")
25 tab_women <- res_women %>%
26   kable(booktabs = TRUE, caption = " Brier Results for Women Model") %>%
27   kableExtra::kable_styling(font_size = 8, latex_options = "HOLD_position")
28   %>% add_footnote(paste("NHANES Brier score is", round(nhanes_women_
29     brier,4)))
30
31 kable(list(tab_men, tab_women))
32 c(brier_men_fram,brier_women_fram)
33 tab_men

```