Transportability Analysis on Cardiovascular Risk Prediction Model on Real vs. Simulated Target Population

Angel Zheng

Dec 2023 for PHP2550

Abstract

This report conducts a transportability analysis of a cardiovascular risk prediction model, evaluating its performance when applied across different populations. Utilizing data from the Framingham Heart Study and NHANES 2017, the analysis focuses on comparing model performances between the original population, a real target population, and a simulated population mimicking the statistical attributes of the target. Exploratory data analysis and model implementation reveal insights into the model's generalizability and effectiveness in diverse populations. Through simulations and statistical modeling, the report examines the model's behavior in simulated datasets, providing insights into its performance in varying population contexts.

1 Introduction

In both clinical practice and research, the construction of predictive models using data from one population and applying them to others is a common practice. The overarching goal is to ensure the reliability and effectiveness of these models when transitioning from their original source population to diverse target populations, a critical aspect referred to as 'generalizability.'

However, achieving this generalizability can be complex due to differences in the distributions of source and target populations, posing challenges to the effective transfer of models. This report delves into this challenge through a comprehensive 'transportability analysis.' It aims to thoroughly examine a model developed within a source population and its subsequent application to a distinct target population, meticulously considering their unique distribution patterns and characteristics.

Central to this analysis is the comparison between simulation-based and data-based approaches. In addition to studying a real target population, this research includes an in-depth examination of a simulated population. This simulated population is deliberately crafted, mirroring the statistical characteristics observed in the source population while aligning with the distribution patterns of the intended target population. This deliberate inclusion enables a meticulous evaluation of how well the model performs when confronted with a population that closely mimics the statistical attributes of the desired target, thereby emphasizing the comparative analysis between simulation and data-based methodologies.

2 Exploratory Data Analysis

Framingham Study

[1]The Framingham Heart Study, launched in 1948, pioneered cardiovascular disease research by tracking 5,209 participants in Framingham, Massachusetts. Conducted biennially, this study highlighted risk factors and disease markers, including blood pressure, chemistry, lung function, habits, and medication use.

Examining events like Angina Pectoris, Myocardial Infarction, Heart Failure, and Cerebrovascular disease, it observed participants' health for 24 years. The dataset provided contains clinic, lab, questionnaire, and event data from 4,434 participants, collected over three periods from 1956 to 1968.

Preprocessing is done to subset of this study data: selects certain covariates: Sex `SEX`, Serum Total Cholesterol `TOTCHOL`, Age `AGE`, Systolic Blood Pressure `SYSBP`, Diastolic Blood Pressure `DIABP`, Current cigarette smoking at exam `CURSMOKE`, Diabetic `DIABETES`, Use of Anti-hypertensive medication at exam `BPMEDS`, High Density Lipoprotein Cholesterol `HDLC`, BMI `BMI` (More information on the complete covariates collected could be found in the reference); removes non-completed and censored rows; splits `SYSBP` to two additional columns `SYSBP_UT` and `SYSBP_T` corresponding to the Systolic Blood Pressure for subjects not on hypertension medication and on medication respectively. The result dataset is being used as the source population data.

Table summary and Histograms are generated below. For histograms, only continuous covariates are plotted.

Table 1: Summary Statistics of Framingham Data

	1	2	p	test
n	1094	1445		
CVD (mean (SD))	0.33 (0.47)	0.17 (0.37)	<0.001	
SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001	
TOTCHOL (mean (SD))	226.44 (41.49)	246.32 (45.51)	<0.001	
AGE (mean (SD))	60.01 (8.18)	60.55 (8.40)	0.106	
SYSBP (mean (SD))	138.94 (20.89)	139.94 (23.71)	0.272	
DIABP (mean (SD))	81.99 (11.31)	80.34 (11.06)	<0.001	
CURSMOKE (mean (SD))	0.39 (0.49)	0.31 (0.46)	<0.001	
DIABETES (mean (SD))	0.09 (0.28)	0.07 (0.25)	0.037	
BPMEDS (mean (SD))	0.11 (0.32)	0.18 (0.38)	<0.001	
HDLC (mean (SD))	43.63 (13.37)	53.07 (15.67)	<0.001	
BMI (mean (SD))	26.25 (3.47)	25.55 (4.22)	<0.001	
SYSBP_UT (mean (SD))	121.04 (46.69)	111.49 (55.89)	<0.001	
SYSBP_T (mean (SD))	17.90 (50.93)	28.45 (61.53)	<0.001	

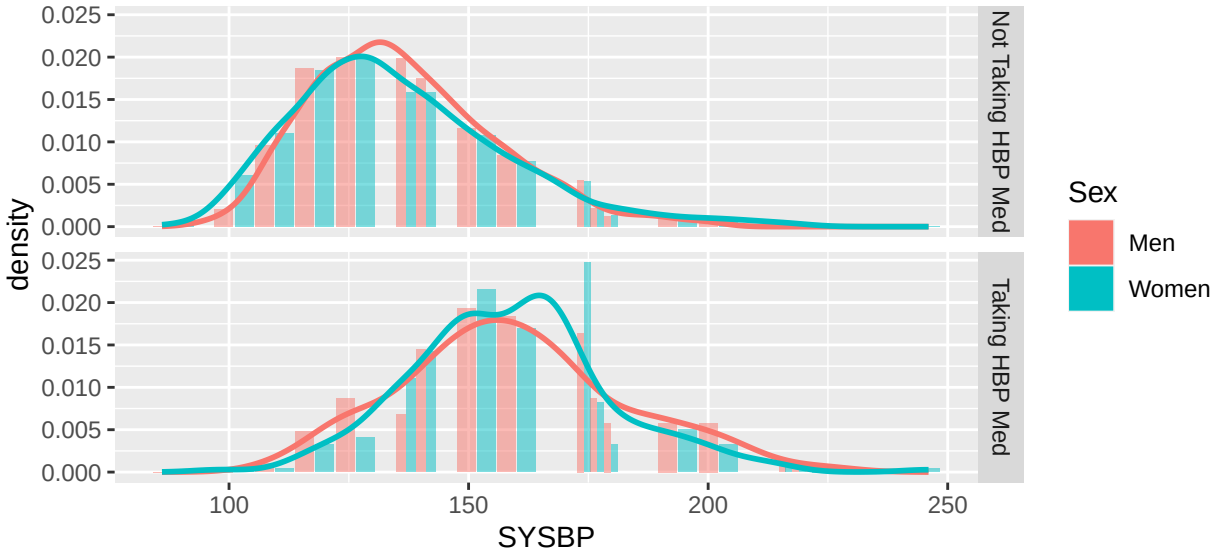

Figure 1: Histograms of Framingham SYSBP by Sex and BPMEDS Status

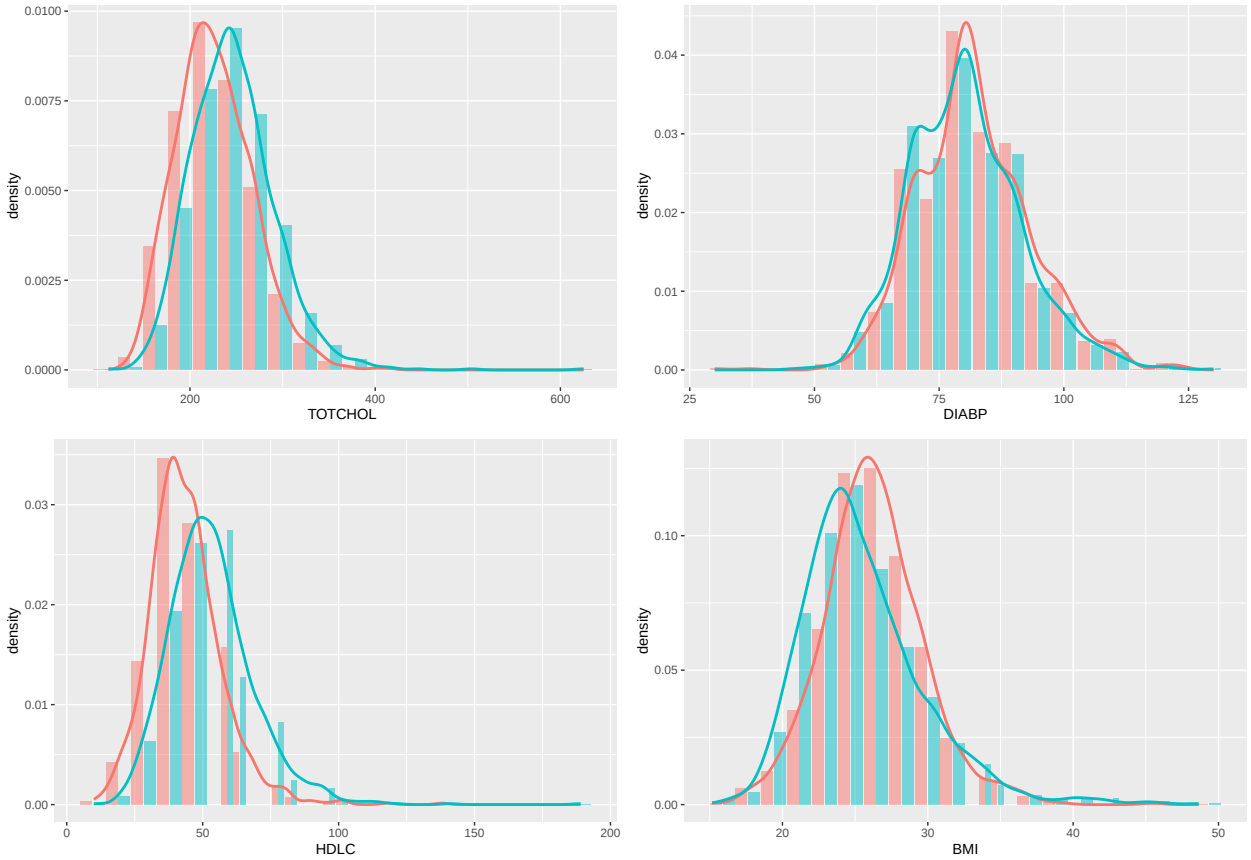

Figure 2: Histograms of Other Framingham Continuous Covariates by Sex

NHANES Study

nhanesA was created to provide versatile access for retrieving data from the National Health and Nutrition Examination Survey (NHANES), which is overseen by the National Center for Health Statistics (NCHS) and offers public accessibility. Starting in 1999, the NHANES survey has been consistently conducted every two years, commencing with the initial period in 1999-2000.[2] This specific report utilizes data exclusively from the 2017 survey. Covariates are selected based on those included in Framingham study. Similarly, non-completed rows are excluded, and `SYSBP` column is splitted to two not on and on hypertension medication.

The NHANES 2017 data that's retrieved is being set to the target population data in this report, and is being combined with the Framingham dataset to create a composite dataset for further analysis described in the Method section.

Table summary and Histograms are generated below. For histograms, only continuous covariates are plotted.

Table 2: Missingness Summary

Variable Name	Number of Missing	Percentage % of Missing
CURSMOKE	3398	36.719257
BPMEDS	3398	36.719257
SYSBP	2952	31.899719
DIABP	2952	31.899719
HDLC	2516	27.188243
TOTCHOL	2516	27.188243
BMI	1249	13.496866
DIABETES	361	3.901016

Table 3: Summary Statistics of Framingham Data

	1	2	p	test
n	2105	2205		
SEQN (mean (SD))	98306.09 (2714.29)	98285.62 (2686.95)	0.804	
SYSBP (mean (SD))	126.44 (16.83)	123.70 (20.36)	<0.001	
DIABP (mean (SD))	73.05 (12.48)	69.58 (13.65)	<0.001	
SEX (mean (SD))	1.00 (0.00)	2.00 (0.00)	<0.001	
AGE (mean (SD))	50.15 (18.83)	48.90 (18.57)	0.029	
BMI (mean (SD))	29.19 (6.25)	29.84 (7.96)	0.003	
HDLC (mean (SD))	48.11 (13.59)	58.10 (15.68)	<0.001	
CURSMOKE (mean (SD))	0.20 (0.40)	0.14 (0.35)	<0.001	
BPMEDS (mean (SD))	0.30 (0.46)	0.29 (0.45)	0.584	
TOTCHOL (mean (SD))	183.10 (41.65)	190.51 (41.20)	<0.001	
DIABETES (mean (SD))	0.18 (0.38)	0.12 (0.33)	<0.001	
SYSBP_UT (mean (SD))	86.46 (57.73)	83.64 (55.42)	0.101	
SYSBP_T (mean (SD))	39.98 (62.19)	40.07 (63.62)	0.964	

3 Method

I Source Data (Framingham)

First part in this transportability analysis is on the source population itself. In this section, [3]a model built from the paper *General Cardiovascular Risk Profile for Use in Primary Care* is implemented. To increase

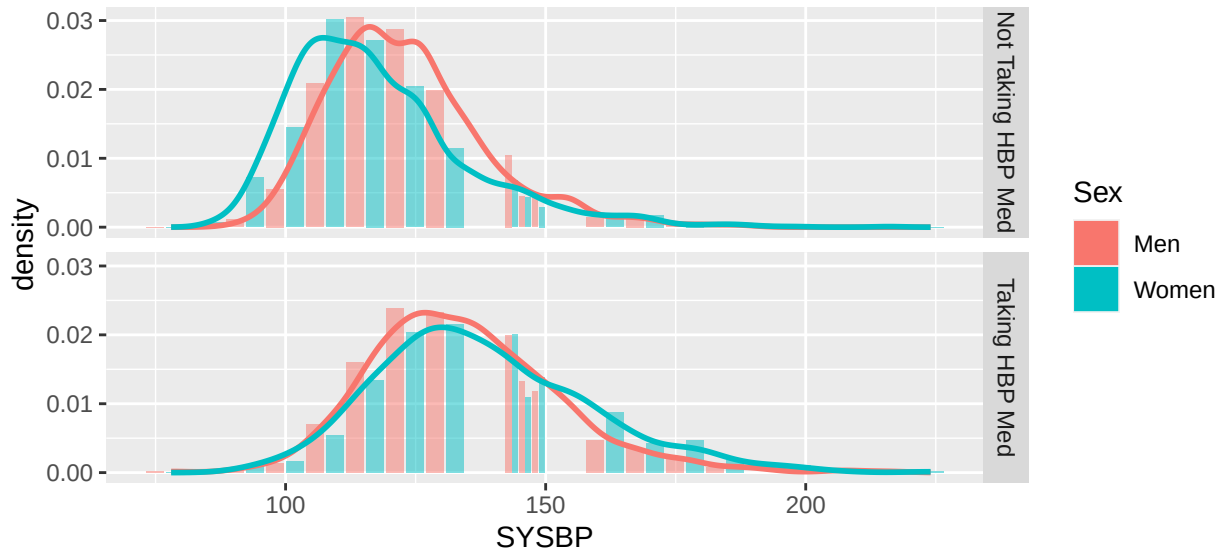

Figure 3: Histograms of NHANES 2017 SYSBP by Sex and BPMEDS Status

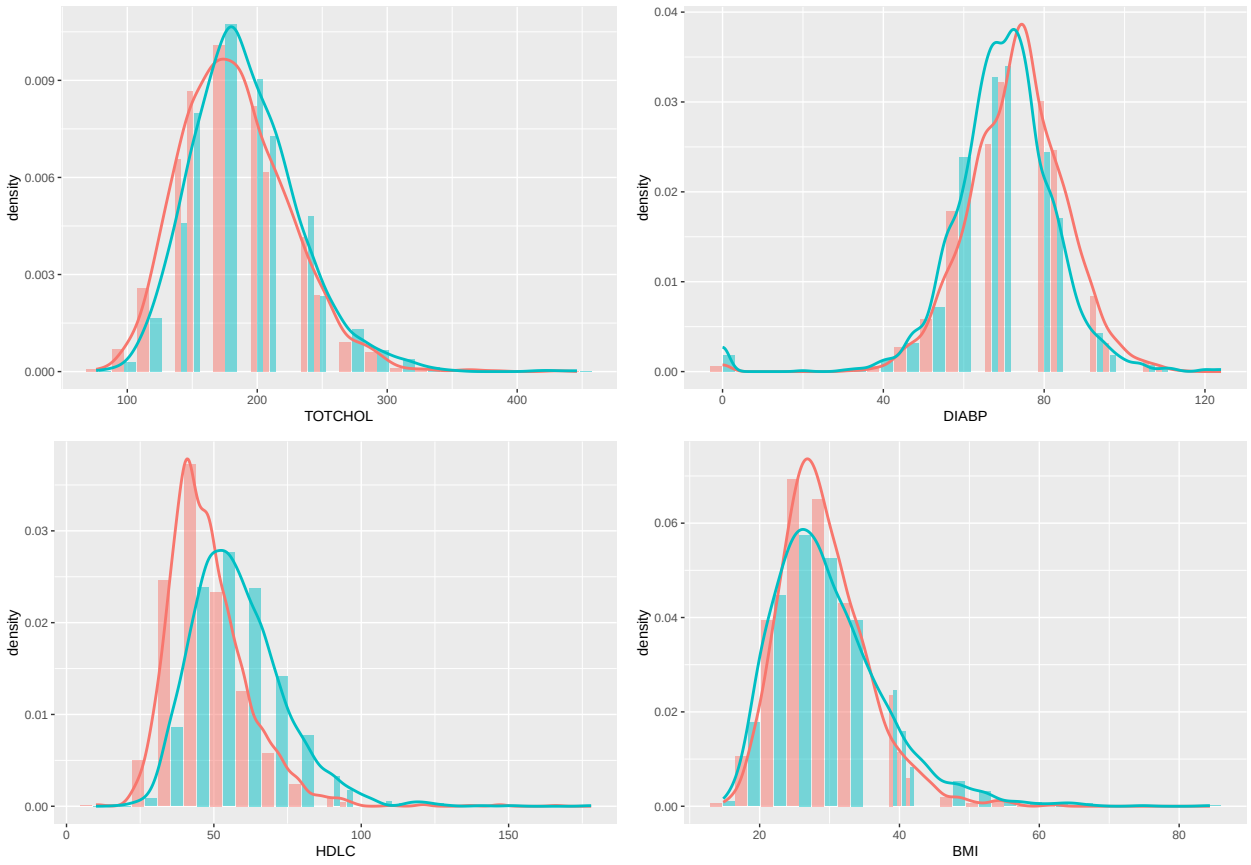

Figure 4: Histograms of Other NHANES 2017 Continuous Covariates by Sex

robustness, the source data is test split with 70% train data for use of model implementing, and 30% test data to evaluate model performance. The model is consist of two models of exact same covariates, just one for each sex.

model $g_{\widehat{\beta}}(X_i)$: CVD $\sim$ log(HDLC) + log(TOTCHOL) + log(AGE) + log(SYSBP_UT + 1) + log(SYSBP_T + 1) + CURSMOKE + DIABETES

where:

- X_i represents all covariates
- `CVD` represents the binary response variable Cardiovascular Disease Status,
- `HDLC`, `TOTCHOL`, `SYSBP_UT`, `SYSBP_T` are continuous predictor variables and are log-transformed,
- `CURSMOKE`, `DIABETES` are categorical predictor variables,
- `+1` inside the logarithm functions is used to handle cases where covariate might contain zero values.

II Composite Data

The composite data is just combining Framingham and NHANES 2017 datasets with their common covariates. The combined dataset is also test-splitted into 70% train data and 30% test data. The same prediction model g is implemented, but this time on the framingham part of train data. To combine the two sets, an indicator `STUDY` or `S` is assigned to each dataset. `S` = 1 indicates data coming from Framingham source population, and `S` = 0 indicates data coming from NHANES 2017 target population.

The formula for transportability analysis is:

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

where $\widehat{o}(X)$ is an estimator for the inverse-odds weights in the test set, $\frac{\Pr[S=0|X, D_{\text{test}}=1]}{\Pr[S=1|X, D_{\text{test}}=1]}$. To obtain these inverse-odds weights, a logistic model for indicator `S` is fitted using the exact same covariates as those in predictive model g.

III Simulation

Simulating a dataset that follows distributions of Framingham data but mimics NHANES 2017 needs information gathered from the Exploratory Data Analysis. In this simulation, the number of iteration is set to 100 based on the formula for Monte Carlo SE of bias. Setting number of simulation to 100 can achieve a SE of 0.02, assuming $SD(\theta) \leq 0.02$. Then, each of the 100 simulated NHANES dataset is performed same procedure as in the Composite Data Method Section to calculate estimator for Brier risk score. The 100 estimated Brier risk scores are averaged out to get the final Brier risk score for model performance on simulation dataset.

$$\text{Monte Carlo SE(Bias)} = \sqrt{\text{Var}(\hat{\theta})/n_{\text{sim}}} = \sqrt{\frac{0.02^2}{100}} = 0.002$$

Univariate: Continuous and Binary Variables

As seen in the histograms for continuous variables in Framingham population, they roughly follow normal distribution with all values being non-negative. A function `rtruncnorm` is used to simulate truncunated continuous variables that follows normal distribution. Then extracting the mean and standard deviation

from NHANES 2017 summary table, the result simulated variable would have the similar mean and sd with those of NHANES 2017 population while following distributions of Framingham population.

One thing to emphasize in the simulation is the manually-splitted variables `SYSBP_UT` and `SYSBP_T`: Although the continuous variable `SYSBP` follows normal distribution, the splitted variables contains values of zero when the hypertension medication status is opposite to setting of that variable (i.e. when `SYSBP_UT` has a value, it means `BPMEDS` = 0 and thus the correspond `SYSBP_T` value is 0). Therefore, the mean and sd being input into `rtruncnorm` for these two variables are based on the corrected summary statistics below instead of values from the overall summary statistics.

The only variable in model that is not normal distribution is `AGE`, and that follows uniform distribution. In this simulation, it is assumed that within the given range each age is equally likely to be included to study population. Although the range of age is not given in the overall summary table, it is usually available in description of study population, recruit process, or inclusion criteria.

Table 4: The Correct Summary Statistics for SYSBP by HPD Med

BPMEDS	SEX	mean.SYSBP_UT	sd.SYSBP_UT	mean.SYSBP_T	sd.SYSBP_T
0	1	123.14	15.11	0.00	0.00
0	2	117.84	17.20	0.00	0.00
1	1	0.00	0.00	134.23	18.08
1	2	0.00	0.00	138.05	20.38

The binary variables included in the predictive model g are `DIABETES` and `CURSMOKE`, but in the simulation here `BPMEDS` and `SEX` are also simulated in order to assign the simulated continuous covariates to their correspond `BPMEDS` status and sex. To simulate binary variables, the total number of 1's and total number of 0's for each binary variable from the NHANES 2017 data are cloned, but the order of those binary responses shuffles using `sample` function.

Multivariate: Framingham correlation

Multivariate: NHANES 2017 correlation

4 Result and Discussion

I Source Data (Framingham)

Table 5: Model Evaluation Metrics on Framingham

	Men	Women
Sensitivity	0.3877551	0.1250000
Specificity	0.8973214	0.9890110
Brier Score	0.1703080	0.1204236
AUC	0.7783345	0.7957418

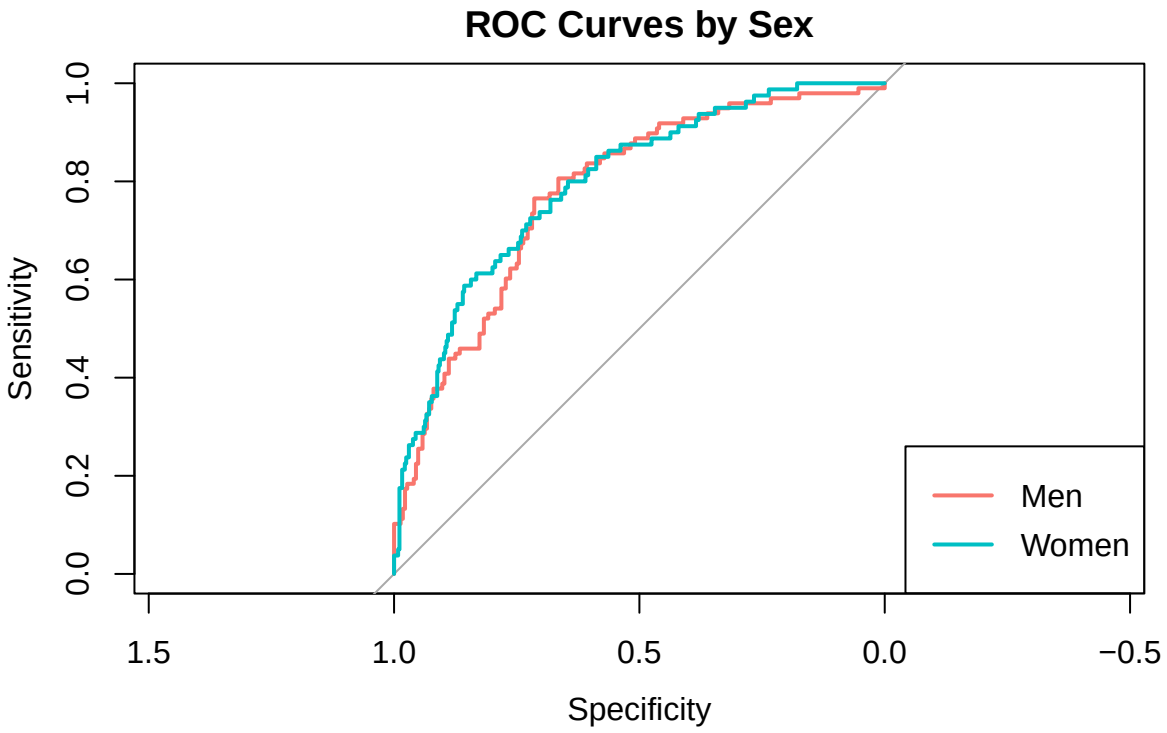

Figure 5: ROC Curves for Model Performance on Framingham

II, III Composite Data and Simulation Data

Table 6: Brier Risk Score Estimation for Model Performance in Five Settings

	Men	Women
Framingham	0.1703080	0.1204236
Framingham + NHANES 2017	0.0944621	0.0562791
+ Univariate Simulation	0.1712567	0.0560969
+ Multivariate, Fram correlation	0.0000000	0.0000000
+ Multivariate, NHANES correlation	0.0000000	0.0000000

The table summarizes the estimate of Brier score in all five settings. It can be seen that for first three settings, the model performance yields a lower Brier score in women than in men, signifies that the model implemented with women population of source Framingham data predicts more accurately than with men population. When the transportability analysis is conducted using composite dataset for Framingham population and real NHANES 2017 population, the brier score gets smaller for both men and women models compared to the source setting. Furthermore, when the transportability analysis is conducted again using composite dataset for Framingham population and simulated NHANES population, the brier score is large in men but small in women. As seen in the histograms, some of the covairates have skewed distributions in NHANES 2017 data. The decrease in brier score could be due to that the simulated target population without skewness has a more similar patterns to the source population.

The latter two simulation settings taking into consideration of the relationship across variables. However, the brier score becomes too small to display in those two settings.

The transportability analysis has strength that it allows examination on generalibility of model, and in this report specifically, the model generalizes relatively accurately onto real and simulated target population. However, its concerns arise in two aspects. Firstly, there is missingness in the target population that could not be determined missingness type. Therefore, multiple imputation is not used in this report, and the summary statistics used to generate simulation based solely on the complete rows from NHANES 2017 population. Secondly, the simulation of covariates only takes into consideration the distributions themselves, but there could have underlying intercorrelation between certain covariates. The correlation relationship, if present, is lost in process of simulate distributions separately for each covariate, and thus might bias the estimates.

5 Conclusion

In conclusion, the transportability analysis conducted demonstrates the source population model predicts more accurately in the real target population. When the target population is simulated based on distributions of source population and summary statistics of real target population, the model predicts even more accurately. However, this pattern could be exclusively to the Framingham and NHANES 2017 populations. Whether a model from source population would always fit simulated target population better than real target population cannot be concluded in general. This model specifically has descent generalizability and is well-transported to broader population.

6 Reference

[1] Framingham Heart Study. (n.d.). RiskCommunicator Package. Retrieved from https://search.r-project.org/CRAN/refmans/riskCommunicator/html/framingham.html

[2] NHANES 2017. (n.d.). Introducing nhanesA Package. Retrieved from https://cran.r-project.org/web/packages/nhanesA/vignettes/Introducing_nhanesA.html

[3] D'Agostino, R. B., Sr, Vasan, R. S., Pencina, M. J., Wolf, P. A., Cobain, M., Massaro, J. M., & Kannel, W. B. (2008). General cardiovascular risk profile for use in primary care: the Framingham Heart Study. Circulation, 117(6), 743–753. https://doi.org/10.1161/CIRCULATIONAHA.107.699579

7 Code Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning=FALSE)
library(riskCommunicator)
library(tidyverse)
library(dplyr)
library(tableone)
library(ggplot2)
library(gridExtra)
library(truncnorm)
library(nhanesA)
library(kableExtra)
library(pROC)
library(naniar)
library(faux)
# load the source dataset
data("framingham")

# The Framingham data has been used to create models for cardiovascular risk.
# The variable selection and model below are designed to mimic the models used
# in the paper General Cardiovascular Risk Profile for Use in Primary Care

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

kableone(CreateTableOne(data=framingham_df, strata = c("SEX")),
         booktabs = TRUE, caption = "Summary Statistics of Framingham Data") %>%
  kableExtra::kable_styling(font_size = 8, latex_options = c("repeat_header", "HOLD_position"))

# helper function for histogram + density plots
hist_lines <- function(df, col) {
  ggplot(data = df, aes(x=.data[[col]])) +
    geom_histogram(aes(y=..density.., fill=as.factor(SEX)), bins = 20, position="dodge2", alpha=0.5)+
    geom_density(size=1, aes(color=as.factor(SEX))) +
    theme(legend.position = "none") +
    xlab(col)
}
```

```r
# density histograms for Framingham SYSBP by sex and by BPMEDS
SEX.labs <- c("Men", "Women")
names(SEX.labs) <- c("1", "2")

BPMEDS.labs <- c("Not Taking HBP Med", "Taking HBP Med")
names(BPMEDS.labs) <- c("0", "1")

ggplot(data = framingham_df, aes(x=framingham_df[["SYSBP"]])) +
  geom_histogram(aes(y=..density.., fill=as.factor(SEX)), bins = 20, position="dodge2", alpha=0.5)+
  geom_density(size=1, aes(color=as.factor(SEX))) +
  facet_grid(BPMEDS~., labeller = labeller(BPMEDS = BPMEDS.labs)) +
  xlab("SYSBP") +
  scale_color_discrete(name = "Sex", labels = c("Men", "Women")) +
  scale_fill_discrete(name = "Sex", labels = c("Men", "Women"))


# density histograms for other continuous covariates
hist_lines(framingham_df, "TOTCHOL")
hist_lines(framingham_df, "DIABP")
hist_lines(framingham_df, "HDLC")
hist_lines(framingham_df,"BMI")

# blood pressure, demographic, bmi, smoking, and hypertension info
bpx_2017 <- nhanes("BPX_J") %>%
  select(SEQN, BPXSY1, BPXDI1) %>%
  rename(SYSBP = BPXSY1, DIABP = BPXDI1)
demo_2017 <- nhanes("DEMO_J") %>%
  select(SEQN, RIAGENDR, RIDAGEYR) %>%
  rename(SEX = RIAGENDR, AGE = RIDAGEYR)
bmx_2017 <- nhanes("BMX_J") %>%
  select(SEQN, BMXBMI) %>%
  rename(BMI = BMXBMI)
smq_2017 <- nhanes("SMQ_J") %>%
  mutate(CURSMOKE = case_when(SMQ040 %in% c(1,2) ~ 1,
                              SMQ040 == 3 ~ 0,
                              SMQ020 == 2 ~ 0)) %>%
  select(SEQN, CURSMOKE)
bpq_2017 <- nhanes("BPQ_J") %>%
  mutate(BPMEDS = case_when(
    BPQ020 == 2 ~ 0,
    BPQ040A == 2 ~ 0,
    BPQ050A == 1 ~ 1,
    TRUE ~ NA )) %>%
  select(SEQN, BPMEDS)
tchol_2017 <- nhanes("TCHOL_J") %>%
  select(SEQN, LBXTC) %>%
  rename(TOTCHOL = LBXTC)
hdl_2017 <- nhanes("HDL_J") %>%
  select(SEQN, LBDHDD) %>%
  rename(HDLC = LBDHDD)
diq_2017 <- nhanes("DIQ_J") %>%
  mutate(DIABETES = case_when(DIQ010 == 1 ~ 1,
                              DIQ010 %in% c(2,3) ~ 0,
```

```
                                TRUE ~ NA)) %>%
  select(SEQN, DIABETES)

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

# 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)

missingness_NHANES <-
  df_2017 %>%
  miss_var_summary() %>%
  filter(n_miss > 0)

colnames(missingness_NHANES) <- c("Variable Name", "Number of Missing", "Percentage % of Missing")

missingness_NHANES %>%
  slice(-c(1:2)) %>%
  kable(booktabs = TRUE, caption = "Missingness Summary") %>%
  kableExtra::kable_styling(font_size = 8, latex_options = c("repeat_header", "HOLD_position"))
# use only complete target population data
df_2017 <- na.omit(df_2017)

kableone(CreateTableOne(data = df_2017, strata = c("SEX")),
         booktabs = TRUE, caption = "Summary Statistics of Framingham Data") %>%
  kableExtra::kable_styling(font_size = 8, latex_options = c("repeat_header", "HOLD_position"))

# density histograms for Framingham SYSBP by sex and by BPMEDS
SEX.labs <- c("Men", "Women")
names(SEX.labs) <- c("1", "2")

BPMEDS.labs <- c("Not Taking HBP Med", "Taking HBP Med")
names(BPMEDS.labs) <- c("0", "1")

ggplot(data = df_2017, aes(x=df_2017[["SYSBP"]])) +
  geom_histogram(aes(y=..density.., fill=as.factor(SEX)), bins = 20, position="dodge2", alpha=0.5)+
  geom_density(size=1, aes(color=as.factor(SEX))) +
  facet_grid(BPMEDS~., labeller = labeller(BPMEDS = BPMEDS.labs)) +
  xlab("SYSBP") +
  scale_color_discrete(name = "Sex", labels = c("Men", "Women")) +
  scale_fill_discrete(name = "Sex", labels = c("Men", "Women"))
# density histograms for other continuous covariates
hist_lines(df_2017, "TOTCHOL")
```

```r
hist_lines(df_2017, "DIABP")
hist_lines(df_2017, "HDLC")
hist_lines(df_2017,"BMI")

# test split the framingham dataset
set.seed(1)
fram_test <- sample(c(TRUE, FALSE), size = nrow(framingham_df), replace = TRUE, prob = c(0.3, 0.7))
framingham_df_train <- framingham_df[-fram_test,]
framingham_df_test <- framingham_df[fram_test,]

# factor variables
framingham_df_train$CURSMOKE <- as.factor(framingham_df_train$CURSMOKE)
framingham_df_train$DIABETES <- as.factor(framingham_df_train$DIABETES)
framingham_df_test$CURSMOKE <- as.factor(framingham_df_test$CURSMOKE)
framingham_df_test$DIABETES <- as.factor(framingham_df_test$DIABETES)
# Filter train and test set to each sex
framingham_df_train_men <- framingham_df_train %>% filter(SEX == 1)
framingham_df_train_women <- framingham_df_train %>% filter(SEX == 2)

framingham_df_test_men <- framingham_df_test %>% filter(SEX == 1)
framingham_df_test_women <- framingham_df_test %>% filter(SEX == 2)

# prediction model
mod_men <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
      data= framingham_df_train_men, family= "binomial")


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

# store number of subjects from each study
n1 <- nrow(framingham_df)
n0 <- nrow(df_2017)

# re-order the columns in nhanes to match framingham
df_2017 <- df_2017 %>%
  mutate(CVD = rep(NA, n0)) %>%
  select(c(14, 4, 10, 5, 2, 3, 8, 11, 9, 7, 6, 12, 13))

# indicator S for study
framingham_df$STUDY <- rep(1, n1)
df_2017$STUDY <- rep(0, n0)

# create combined dataset
combine <- rbind(framingham_df, df_2017)
combine[,c(2, 7:9, 14)] <- lapply(combine[,c(2, 7:9, 14)], as.factor)

# new test set: NHANES 2017 added
framingham_df_train$STUDY <- rep(1, nrow(framingham_df_train))
framingham_df_test$STUDY <- rep(1, nrow(framingham_df_test))
```

```r
combine_train <- framingham_df_train
combine_test <- rbind(framingham_df_test, df_2017)
# separate combined train set by sex
combine_train_men <- combine_train %>% filter(SEX == 1)
combine_train_women <- combine_train %>% filter(SEX == 2)

# separate combined test set by sex
combine_test_men <- combine_test %>% filter(SEX == 1)
combine_test_women <- combine_test %>% filter(SEX == 2)

# logistic model for indicator S
mod.S_men <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_men, family = "binomial")

mod.S_women <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_women, family = "binomial")

# add a column for predicted probability of S=1
combine_test$STUDY1.pred <- rep(NA, nrow(combine_test))
combine_test[combine_test$SEX == 1,]$STUDY1.pred <-
  predict(mod.S_men, combine_test[combine_test$SEX == 1,], type = "response")
combine_test[combine_test$SEX == 2,]$STUDY1.pred <-
  predict(mod.S_women, combine_test[combine_test$SEX == 2,], type = "response")

# Fit models with log transforms for all continuous variables
mod_men_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_men[combine_train_men$STUDY == 1,], family= "binomial")

mod_women_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_women[combine_train_women$STUDY == 1,], family= "binomial")

# add a column for predicted outcome CVD
combine_test$CVD.pred <- rep(NA, nrow(combine_test))
combine_test[combine_test$STUDY == 1 & combine_test$SEX == 1,]$CVD.pred <-
  predict(mod_men_2, combine_test[combine_test$STUDY == 1 & combine_test$SEX == 1,], type = "response")
combine_test[combine_test$STUDY == 1 & combine_test$SEX == 2,]$CVD.pred <-
  predict(mod_women_2, combine_test[combine_test$STUDY == 1 & combine_test$SEX == 2,], type = "response

# store number of rows for simulation use
n0_men <- nrow(df_2017[df_2017$SEX == 1,])
n0_women <- nrow(df_2017[df_2017$SEX == 2,])

n_SYSBP_UT_men <- sum(df_2017$BPMEDS == 0 & df_2017$SEX == 1)
n_SYSBP_UT_women <- sum(df_2017$BPMEDS == 0 & df_2017$SEX == 2)

n_SYSBP_T_men <- sum(df_2017$BPMEDS == 1 & df_2017$SEX == 1)
n_SYSBP_T_women <- sum(df_2017$BPMEDS == 1 & df_2017$SEX == 2)

# correcting mean and sd for SYSBP_UT and SYSBP_T
df_2017 %>%
  group_by(BPMEDS, SEX) %>%
```

```
  select(SYSBP_UT, SYSBP_T) %>%
  summarize(mean.SYSBP_UT = round(mean(SYSBP_UT),2),
            sd.SYSBP_UT = round(sd(SYSBP_UT),2),
            mean.SYSBP_T = round(mean(SYSBP_T),2),
            sd.SYSBP_T = round(sd(SYSBP_T),2)) %>%
  kable(caption = "The Correct Summary Statistics for SYSBP by HPD Med", align = "c",booktabs = T) %>%
  kable_styling(full_width=T,latex_options = c('HOLD_position'),font_size = 10)

# empty vectors to be filled in for loop
brier_sim_men <- rep(NA, 100)
brier_sim_women <- rep(NA, 100)


# simulate a new dataset 100 times
for (i in 1:100) {
set.seed(i)
HDLC.sim_men <- rtruncnorm(n0_men, a = 0, mean = 48.11, sd = 13.59)
HDLC.sim_women <- rtruncnorm(n0_women, a = 0, mean = 58.10, sd = 15.68)

TOTCHOL.sim_men <- rtruncnorm(n0_men, a = 0, mean = 183.10, sd = 41.65)
TOTCHOL.sim_women <- rtruncnorm(n0_women, a = 0, mean = 190.51, sd = 41.20)

AGE.sim_men <- sample(18:80, n0_men, replace = T)
AGE.sim_men <- AGE.sim_men * 50.15/mean(AGE.sim_men)
AGE.sim_men <- pmin(pmax(round(AGE.sim_men), 18), 80)

AGE.sim_women <- sample(18:80, n0_women, replace = T)
AGE.sim_women <- AGE.sim_women * 48.90/mean(AGE.sim_women)
AGE.sim_women <- pmin(pmax(round(AGE.sim_women), 18), 80)

SYSBP_UT.sim_men <- rtruncnorm(n_SYSBP_UT_men, a=0, mean = 123.14, sd = 15.11)
SYSBP_UT.sim_women <- rtruncnorm(n_SYSBP_UT_women, a=0, mean = 117.84, sd = 17.20)

SYSBP_T.sim_men <- rtruncnorm(n_SYSBP_T_men, a=0, mean = 134.23, sd = 18.08)
SYSBP_T.sim_women <- rtruncnorm(n_SYSBP_T_women, a=0, mean = 138.05, sd = 20.38)

BPMEDS.sim_men <- c(rep(1, n_SYSBP_T_men), rep(0, n_SYSBP_UT_men))
BPMEDS.sim_men <- sample(BPMEDS.sim_men)
BPMEDS.sim_women <- c(rep(1, n_SYSBP_T_women), rep(0, n_SYSBP_UT_women))
BPMEDS.sim_women <- sample(BPMEDS.sim_women)

CURSMOKE.sim_men <- rbinom(n0_men, size=1, prob = 0.20)
CURSMOKE.sim_women <- rbinom(n0_women, size=1, prob = 0.14)

DIABETES.sim_men <- rbinom(n0_men, size=1, prob = 0.18)
DIABETES.sim_women <- rbinom(n0_women, size=1, prob = 0.12)

SEX.sim <- c(rep(1, n0_men), rep(2, n0_women))
SEX.sim <- sample(SEX.sim)

df_2017.sim <- data.frame(CVD = rep(NA, n0),
                          SEX = as.factor(SEX.sim),
                          STUDY = rep(0, n0))
```

```r
df_2017.sim$TOTCHOL[df_2017.sim$SEX == 1] <- TOTCHOL.sim_men
df_2017.sim$TOTCHOL[df_2017.sim$SEX == 2] <- TOTCHOL.sim_women

df_2017.sim$AGE[df_2017.sim$SEX == 1] <- AGE.sim_men
df_2017.sim$AGE[df_2017.sim$SEX == 2] <- AGE.sim_women

df_2017.sim$DIABP[df_2017.sim$SEX == 1] <- DIABETES.sim_men
df_2017.sim$DIABP[df_2017.sim$SEX == 2] <- DIABETES.sim_women

df_2017.sim$CURSMOKE[df_2017.sim$SEX == 1] <- CURSMOKE.sim_men
df_2017.sim$CURSMOKE[df_2017.sim$SEX == 2] <- CURSMOKE.sim_women
df_2017.sim$CURSMOKE <- as.factor(df_2017.sim$CURSMOKE)

df_2017.sim$DIABETES[df_2017.sim$SEX == 1] <- DIABETES.sim_men
df_2017.sim$DIABETES[df_2017.sim$SEX == 2] <- DIABETES.sim_women
df_2017.sim$DIABETES <- as.factor(df_2017.sim$DIABETES)

df_2017.sim$BPMEDS[df_2017.sim$SEX == 1] <- BPMEDS.sim_men
df_2017.sim$BPMEDS[df_2017.sim$SEX == 2] <- BPMEDS.sim_women

df_2017.sim$HDLC[df_2017.sim$SEX == 1] <- HDLC.sim_men
df_2017.sim$HDLC[df_2017.sim$SEX == 2] <- HDLC.sim_women

df_2017.sim$SYSBP_UT <- rep(0, n0)
df_2017.sim$SYSBP_UT[df_2017.sim$SEX == 1 & df_2017.sim$BPMEDS == 0] <- SYSBP_UT.sim_men
df_2017.sim$SYSBP_UT[df_2017.sim$SEX == 2 & df_2017.sim$BPMEDS == 0] <- SYSBP_UT.sim_women

df_2017.sim$SYSBP_T <- rep(0, n0)
df_2017.sim$SYSBP_T[df_2017.sim$SEX == 1 & df_2017.sim$BPMEDS == 1] <- SYSBP_T.sim_men
df_2017.sim$SYSBP_T[df_2017.sim$SEX == 2 & df_2017.sim$BPMEDS == 1] <- SYSBP_T.sim_women

df_2017.sim_men <- df_2017.sim[df_2017.sim$SEX == 1,]
df_2017.sim_women <- df_2017.sim[df_2017.sim$SEX == 1,]

# create combined dataset
combine2 <- rbind(subset(framingham_df, select = colnames(df_2017.sim)),
                  df_2017.sim)

# test split the combined dataset
combine_train2 <- framingham_df_train
combine_test2 <- rbind(framingham_df_test[-c(5, 11)], df_2017.sim)


# separate combined train set by sex
combine_train_men2 <- combine_train2 %>% filter(SEX == 1)
combine_train_women2 <- combine_train2 %>% filter(SEX == 2)

# separate combined test set by sex
combine_test_men2 <- combine_test2 %>% filter(SEX == 1)
combine_test_women2 <- combine_test2 %>% filter(SEX == 2)

# logistic model for indicator S
mod.S_men <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
```

```
              log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_men2, family = "binomial")

mod.S_women <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
              log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_women2, family = "binomial")

# add a column for predicted probability of S=1
combine_test2$STUDY1.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$SEX == 1,]$STUDY1.pred <-
  predict(mod.S_men, combine_test2[combine_test2$SEX == 1,], type = "response")
combine_test2[combine_test2$SEX == 2,]$STUDY1.pred <-
  predict(mod.S_women, combine_test2[combine_test2$SEX == 2,], type = "response")


# Fit models with log transforms for all continuous variables
mod_men_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_men2[combine_train_men2$STUDY == 1,], family= "binomial")

mod_women_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_women2[combine_train_women2$STUDY == 1,], family= "binomial")

# add a column for predicted outcome CVD
combine_test2$CVD.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,]$CVD.pred <-
  predict(mod_men_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,], type = "response
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,]$CVD.pred <-
  predict(mod_women_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,], type = "respo

# calculate the estimated brier risk
brier_hat_calculation2 <-
  combine_test2 %>%
  mutate(o = (1-combine_test2$STUDY1.pred)/combine_test2$STUDY1.pred,
         CVD.diff2 = (CVD - CVD.pred)**2) %>%
  group_by(SEX, STUDY) %>%
  summarize(n = n(),
            sum(o * CVD.diff2))

brier_sim_men[i] <- as.numeric(brier_hat_calculation2[2,4]/brier_hat_calculation2[1,3])
brier_sim_women[i] <- as.numeric(brier_hat_calculation2[4,4]/brier_hat_calculation2[3,3])
}

# empty vectors to be filled in for loop
brier_sim2_men <- rep(NA, 100)
brier_sim2_women <- rep(NA, 100)


# simulate a new dataset 100 times
for (i in 1:100) {
set.seed(i)
AGE.sim_men <- sample(18:80, n0_men, replace = T)
AGE.sim_men <- AGE.sim_men * 50.15/mean(AGE.sim_men)
AGE.sim_men <- pmin(pmax(round(AGE.sim_men), 18), 80)
```

```r
AGE.sim_women <- sample(18:80, n0_women, replace = T)
AGE.sim_women <- AGE.sim_women * 48.90/mean(AGE.sim_women)
AGE.sim_women <- pmin(pmax(round(AGE.sim_women), 18), 80)

BPMEDS.sim_men <- c(rep(1, n_SYSBP_T_men), rep(0, n_SYSBP_UT_men))
BPMEDS.sim_men <- sample(BPMEDS.sim_men)
BPMEDS.sim_women <- c(rep(1, n_SYSBP_T_women), rep(0, n_SYSBP_UT_women))
BPMEDS.sim_women <- sample(BPMEDS.sim_women)

CURSMOKE.sim_men <- rbinom(n0_men, size=1, prob = 0.20)
CURSMOKE.sim_women <- rbinom(n0_women, size=1, prob = 0.14)

DIABETES.sim_men <- rbinom(n0_men, size=1, prob = 0.18)
DIABETES.sim_women <- rbinom(n0_women, size=1, prob = 0.12)

SEX.sim <- c(rep(1, n0_men), rep(2, n0_women))
SEX.sim <- sample(SEX.sim)

# apply correlation matrix in simulation
cmat_fram <- cor(framingham_df[,c("SYSBP", "TOTCHOL", "DIABP", "HDLC", "BMI")])
df_2017.sim2 <- rnorm_multi(n=n0, 5, 0, 1, cmat_fram, varnames = colnames(cmat_fram))

df_2017.sim2$SEX <- SEX.sim

df_2017.sim2_men <- df_2017.sim2[df_2017.sim2$SEX == 1,]
df_2017.sim2_women <- df_2017.sim2[df_2017.sim2$SEX == 2,]

df_2017.sim2_men$AGE <- AGE.sim_men
df_2017.sim2_women$AGE <- AGE.sim_women

df_2017.sim2_men$CURSMOKE <- CURSMOKE.sim_men
df_2017.sim2_women$CURSMOKE <- CURSMOKE.sim_women

df_2017.sim2_men$DIABETES <- DIABETES.sim_men
df_2017.sim2_women$DIABETES <- DIABETES.sim_women

df_2017.sim2_men$BPMEDS <- BPMEDS.sim_men
df_2017.sim2_women$BPMEDS <- BPMEDS.sim_women

df_2017.sim2_men$SYSBP_UT <- ifelse(df_2017.sim2_men$BPMEDS == 0,
                                    df_2017.sim2_men$SYSBP, 0)
df_2017.sim2_men$SYSBP_T <- ifelse(df_2017.sim2_men$BPMEDS == 1,
                                    df_2017.sim2_men$SYSBP, 0)

df_2017.sim2_women$SYSBP_UT <- ifelse(df_2017.sim2_women$BPMEDS == 0,
                                    df_2017.sim2_women$SYSBP, 0)

df_2017.sim2_women$SYSBP_T <- ifelse(df_2017.sim2_women$BPMEDS == 1,
                                    df_2017.sim2_women$SYSBP, 0)

df_2017.sim2_men <- df_2017.sim2_men %>%
  mutate(CVD = rep(NA, n0_men),
         SEX = rep(1, n0_men),
```

```
         STUDY = rep(0, n0_men))

df_2017.sim2_women <- df_2017.sim2_women %>%
  mutate(CVD = rep(NA, n0_women),
         SEX = rep(1, n0_women),
         STUDY = rep(0, n0_women))

# create combined dataset
combine2 <- rbind(subset(framingham_df, select = colnames(df_2017.sim)),
                  df_2017.sim)

combine_train_men2 <- framingham_df_train %>% filter(SEX == 1)
combine_train_women2 <- framingham_df_train %>% filter(SEX == 2)

# separate combined test set by sex
framingham_df_test_men$STUDY <- rep(1, nrow(framingham_df_test_men))
framingham_df_test_women$STUDY <- rep(1, nrow(framingham_df_test_women))

combine_test_men2 <- rbind(framingham_df_test_men, df_2017.sim2_men)
combine_test_women2 <- rbind(framingham_df_test_women, df_2017.sim2_men)

# logistic model for indicator S
mod.S_men <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_men2, family = "binomial")

mod.S_women <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_women2, family = "binomial")

# add a column for predicted probability of S=1
combine_test2$STUDY1.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$SEX == 1,]$STUDY1.pred <-
  predict(mod.S_men, combine_test2[combine_test2$SEX == 1,], type = "response")
combine_test2[combine_test2$SEX == 2,]$STUDY1.pred <-
  predict(mod.S_women, combine_test2[combine_test2$SEX == 2,], type = "response")


# Fit models with log transforms for all continuous variables
mod_men_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_men2[combine_train_men2$STUDY == 1,], family= "binomial")

mod_women_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_women2[combine_train_women2$STUDY == 1,], family= "binomial")

# add a column for predicted outcome CVD
combine_test2$CVD.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,]$CVD.pred <-
  predict(mod_men_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,], type = "respons
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,]$CVD.pred <-
  predict(mod_women_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,], type = "respo

# calculate the estimated brier risk
```

```r
brier_hat_calculation2 <-
  combine_test2 %>%
  mutate(o = (1-combine_test2$STUDY1.pred)/combine_test2$STUDY1.pred,
         CVD.diff2 = (CVD - CVD.pred)**2) %>%
  group_by(SEX, STUDY) %>%
  summarize(n = n(),
            sum(o * CVD.diff2))

brier_sim2_men[i] <- as.numeric(brier_hat_calculation2[2,4]/brier_hat_calculation2[1,3])
brier_sim2_women[i] <- as.numeric(brier_hat_calculation2[4,4]/brier_hat_calculation2[3,3])
}

# empty vectors to be filled in for loop
brier_sim3_men <- rep(NA, 100)
brier_sim3_women <- rep(NA, 100)


# simulate a new dataset 100 times
for (i in 1:100) {
set.seed(i)
AGE.sim_men <- sample(18:80, n0_men, replace = T)
AGE.sim_men <- AGE.sim_men * 50.15/mean(AGE.sim_men)
AGE.sim_men <- pmin(pmax(round(AGE.sim_men), 18), 80)

AGE.sim_women <- sample(18:80, n0_women, replace = T)
AGE.sim_women <- AGE.sim_women * 48.90/mean(AGE.sim_women)
AGE.sim_women <- pmin(pmax(round(AGE.sim_women), 18), 80)

BPMEDS.sim_men <- c(rep(1, n_SYSBP_T_men), rep(0, n_SYSBP_UT_men))
BPMEDS.sim_men <- sample(BPMEDS.sim_men)
BPMEDS.sim_women <- c(rep(1, n_SYSBP_T_women), rep(0, n_SYSBP_UT_women))
BPMEDS.sim_women <- sample(BPMEDS.sim_women)

CURSMOKE.sim_men <- rbinom(n0_men, size=1, prob = 0.20)
CURSMOKE.sim_women <- rbinom(n0_women, size=1, prob = 0.14)

DIABETES.sim_men <- rbinom(n0_men, size=1, prob = 0.18)
DIABETES.sim_women <- rbinom(n0_women, size=1, prob = 0.12)

SEX.sim <- c(rep(1, n0_men), rep(2, n0_women))
SEX.sim <- sample(SEX.sim)

# apply correlation matrix in simulation
cmat_nhanes <- cor(df_2017[,c("SYSBP", "TOTCHOL", "DIABP", "HDLC", "BMI")])
df_2017.sim3 <- rnorm_multi(n=n0, 5, 0, 1, cmat_nhanes, varnames = colnames(cmat_nhanes))

df_2017.sim3$SEX <- SEX.sim

df_2017.sim3_men <- df_2017.sim3[df_2017.sim3$SEX == 1,]
df_2017.sim3_women <- df_2017.sim3[df_2017.sim3$SEX == 2,]


df_2017.sim3_men$AGE <- AGE.sim_men
```

```r
df_2017.sim3_women$AGE <- AGE.sim_women

df_2017.sim3_men$CURSMOKE <- CURSMOKE.sim_men
df_2017.sim3_women$CURSMOKE <- CURSMOKE.sim_women

df_2017.sim3_men$DIABETES <- DIABETES.sim_men
df_2017.sim3_women$DIABETES <- DIABETES.sim_women

df_2017.sim3_men$BPMEDS <- BPMEDS.sim_men
df_2017.sim3_women$BPMEDS <- BPMEDS.sim_women

df_2017.sim3_men$SYSBP_UT <- ifelse(df_2017.sim3_men$BPMEDS == 0,
                                    df_2017.sim3_men$SYSBP, 0)
df_2017.sim3_men$SYSBP_T <- ifelse(df_2017.sim3_men$BPMEDS == 1,
                                    df_2017.sim3_men$SYSBP, 0)

df_2017.sim3_women$SYSBP_UT <- ifelse(df_2017.sim3_women$BPMEDS == 0,
                                    df_2017.sim3_women$SYSBP, 0)

df_2017.sim3_women$SYSBP_T <- ifelse(df_2017.sim3_women$BPMEDS == 1,
                                    df_2017.sim3_women$SYSBP, 0)

df_2017.sim3_men <- df_2017.sim3_men %>%
  mutate(CVD = rep(NA, n0_men),
         SEX = rep(1, n0_men),
         STUDY = rep(0, n0_men))

df_2017.sim3_women <- df_2017.sim3_women %>%
  mutate(CVD = rep(NA, n0_women),
         SEX = rep(1, n0_women),
         STUDY = rep(0, n0_women))

# create combined dataset
combine2 <- rbind(subset(framingham_df, select = colnames(df_2017.sim)),
                  df_2017.sim)

combine_train_men2 <- framingham_df_train %>% filter(SEX == 1)
combine_train_women2 <- framingham_df_train %>% filter(SEX == 2)

# separate combined test set by sex
framingham_df_test_men$STUDY <- rep(1, nrow(framingham_df_test_men))
framingham_df_test_women$STUDY <- rep(1, nrow(framingham_df_test_women))

combine_test_men2 <- rbind(framingham_df_test_men, df_2017.sim3_men)
combine_test_women2 <- rbind(framingham_df_test_women, df_2017.sim3_men)

# logistic model for indicator S
mod.S_men <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_men2, family = "binomial")

mod.S_women <- glm(STUDY ~ log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
               log(SYSBP_T+1)+CURSMOKE+DIABETES , data = combine_test_women2, family = "binomial")
```

```r
# add a column for predicted probability of S=1
combine_test2$STUDY1.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$SEX == 1,]$STUDY1.pred <-
  predict(mod.S_men, combine_test2[combine_test2$SEX == 1,], type = "response")
combine_test2[combine_test2$SEX == 2,]$STUDY1.pred <-
  predict(mod.S_women, combine_test2[combine_test2$SEX == 2,], type = "response")


# Fit models with log transforms for all continuous variables
mod_men_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_men2[combine_train_men2$STUDY == 1,], family= "binomial")

mod_women_2 <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
                 data= combine_train_women2[combine_train_women2$STUDY == 1,], family= "binomial")

# add a column for predicted outcome CVD
combine_test2$CVD.pred <- rep(NA, nrow(combine_test2))
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,]$CVD.pred <-
  predict(mod_men_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 1,], type = "response
combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,]$CVD.pred <-
  predict(mod_women_2, combine_test2[combine_test2$STUDY == 1 & combine_test2$SEX == 2,], type = "respo

# calculate the estimated brier risk
brier_hat_calculation2 <-
  combine_test2 %>%
  mutate(o = (1-combine_test2$STUDY1.pred)/combine_test2$STUDY1.pred,
         CVD.diff2 = (CVD - CVD.pred)**2) %>%
  group_by(SEX, STUDY) %>%
  summarize(n = n(),
            sum(o * CVD.diff2))

brier_sim3_men[i] <- as.numeric(brier_hat_calculation2[2,4]/brier_hat_calculation2[1,3])
brier_sim3_women[i] <- as.numeric(brier_hat_calculation2[4,4]/brier_hat_calculation2[3,3])
}
# evaluation metrics helper function
eval_metrics <- function(pred_prob, actual, threshold = 0.5){
  # predicted probabilities and actual values
  prediction <- ifelse(pred_prob > threshold, 1, 0)
  sensitivity <- sum(prediction == 1 & actual == 1) / sum(actual == 1)
  specificity <- sum(prediction == 0 & actual == 0) / sum(actual == 0)
  BS <- mean( (pred_prob - ifelse(actual == 1, 1, 0))^2 )
  AUC <- as.numeric(roc(actual, pred_prob)$auc)
  return(c(sensitivity, specificity, BS, AUC))
}

# predicted values of g on test set framingham
fram_pred_prob_men <- predict(mod_men, framingham_df_test_men, type = "response")
fram_pred_prob_women <- predict(mod_women, framingham_df_test_women, type = "response")


# put the evaluation metrics into a table
```

```r
metrics_table <- data.frame(eval_metrics(fram_pred_prob_men, framingham_df_test_men$CVD),
                            eval_metrics(fram_pred_prob_women, framingham_df_test_women$CVD))

rownames(metrics_table) <- c("Sensitivity", "Specificity", "Brier Score", "AUC")
colnames(metrics_table) <- c("Men", "Women")

metrics_table %>% kable(caption = "Model Evaluation Metrics on Framingham", align = "c",booktabs = T) %>
kable_styling(full_width=T,latex_options = c('HOLD_position'),font_size = 10)

# roc curves
roc_fram_men <- roc(framingham_df_test_men$CVD, fram_pred_prob_men)
roc_fram_women <- roc(framingham_df_test_women$CVD, fram_pred_prob_women)

plot(roc_fram_men, main = "ROC Curves by Sex", col = "#F8766D", lwd = 2)
plot(roc_fram_women, col = "#00BFC4", add = TRUE, lwd = 2)

legend("bottomright", legend = c("Men", "Women"), col = c("#F8766D", "#00BFC4"), lwd = 2)

# calculate the estimated brier risk
brier_hat_calculation <-
  combine_test %>%
  mutate(o = (1-combine_test$STUDY1.pred)/combine_test$STUDY1.pred,
         CVD.diff2 = (CVD - CVD.pred)**2) %>%
  group_by(SEX, STUDY) %>%
  summarize(n = n(),
            sum(o * CVD.diff2))

brier_com_men <- as.numeric(brier_hat_calculation[2,4]/brier_hat_calculation[1,3])
brier_com_women <- as.numeric(brier_hat_calculation[4,4]/brier_hat_calculation[3,3])
# compare all brier values together
brier_table <- data.frame(matrix(c(metrics_table[3,1], metrics_table[3,2],
                                   brier_com_men, brier_com_women,
                                   mean(brier_sim_men), mean(brier_sim_women),
                                   mean(brier_sim2_men), mean(brier_sim2_women),
                                   mean(brier_sim3_men), mean(brier_sim3_women)), nrow = 5, ncol=2, byr

rownames(brier_table) <- c("Framingham", "Framingham + NHANES 2017", " + Univariate Simulation", " + Mul
colnames(brier_table) <- c("Men", "Women")

brier_table %>%
  kable(caption = "Brier Risk Score Estimation for Model Performance in Five Settings", align = "c",bool
  kable_styling(full_width=T,latex_options = c('HOLD_position'),font_size = 10)
performance <- function(beta1,beta_hat){
  #' We assess the bias, MSE, empirical and model-based standard errors for beta_hat
  #' and we will report Monte Carlo standard error
  #' @param beta1, the true beta
  #' @param beta_hat, the estimated beta
  #' @return performance measures
  # get the number of simulations
  n_sim <- length(beta_hat)
  # bias
  bias <- mean(beta_hat)-beta1
  bias.mcse <- sqrt(var(beta_hat)/n_sim)
```

```
  # MSE
  MSE <- mean((beta_hat-beta1)^2)
  MSE.mcse <- sqrt(var((beta_hat-beta1)^2)/n_sim)
  # EmpSE
  EmpSE <- sqrt(var(beta_hat))
  EmpSE.mcse <- EmpSE/sqrt(2*(n_sim-1))
  # power
  power <- sum(beta_hat!=0)/length(beta_hat)
  power.mcse <- sqrt(power*(1-power)/n_sim)
  return(list(Bias=round(bias,3),Bias.se=round(bias.mcse,3),
              MSE=round(MSE,3),MSE.se=round(MSE.mcse,3),
              EmpSE=round(EmpSE,3),EmpSE.se=round(EmpSE.mcse,3),
              Power=round(power,3),Power.se=round(power.mcse,3)))
}
```