Practical Data Analysis Project 3

Tabib Chowdhury

2023-12-05

Abstract

In this collaborative project with Dr. Jon Steingrimsson we evaluate the performance of a model when applied to a different external dataset. Often time prediction models are trained on randomized controlled trials or observational study data which may not represent the demographics of the target population. The Framingham ATP III model has been trained on predominantly white participants and has shown to be limited in terms of its generalization to multiethnic populations.

In this project, we conduct a simulation study where we focus on a risk score model built from the Framingham Heart Study data, and the evaluation will be conducted in the population underlying the National Health and Nutrition Examination Survey (NHANES). To evaluate the NHANES data, we estimate the mean squared error for binary outcome data(i.e brier score) on a target population. This procedure is called transportability analysis. We found that the model evaluates well on the NHANES data achieving an MSE score of around 0.105. After doing this, I simulated new individual level data using the summary statistics of the the NHANES data and proceeded to perform transportability analysis to the simulated data as well. We found that the model performs better on the simulated data compared to the actual NHANES data.

Introduction

Transportability analysis is required for our specific setting because we are attempting to address the critical challenges of assessing the performance of prediction models when applied to populations distinct from those in which they were originally developed. The source data here is the framingham data orginates from a 1948 study where the original goal of the Framingham Heart Study (FHS) was to identify common factors or characteristics that contribute to cardiovascular disease and it's been collected over generations from the city of Framingham, Massachussetts. The issue with this setting is that the city of Framingham, MA is predominantly white and thus it would be hard to generalize this study to a different setting.

In our analysis we calculate the mean squared errors by taking into account the distribution of the covariates from both the source and target data. The target data comes from the NHANES study. The NHANES data set is significantly different from the framingham data. We find that across different settings, the NHANES data performs well on the framingham model.

After evaluating the NHANES data, we move on to simulating our own data using the summary statistics of the NHANES data. To this we follow the ADEMP(aims, data-generating mechanisms, estimands, methods, and performance measures) framework. After simulating a new dataset, we find that the simulated data performs better than the NHANES data on the framingham model.

Exploratory Data Analysis

Before performing any analysis or evaluation, we need to compare the NHANES and framingham data and look at how to deal with missing data. The table below compares the common variables between the two

datasets:

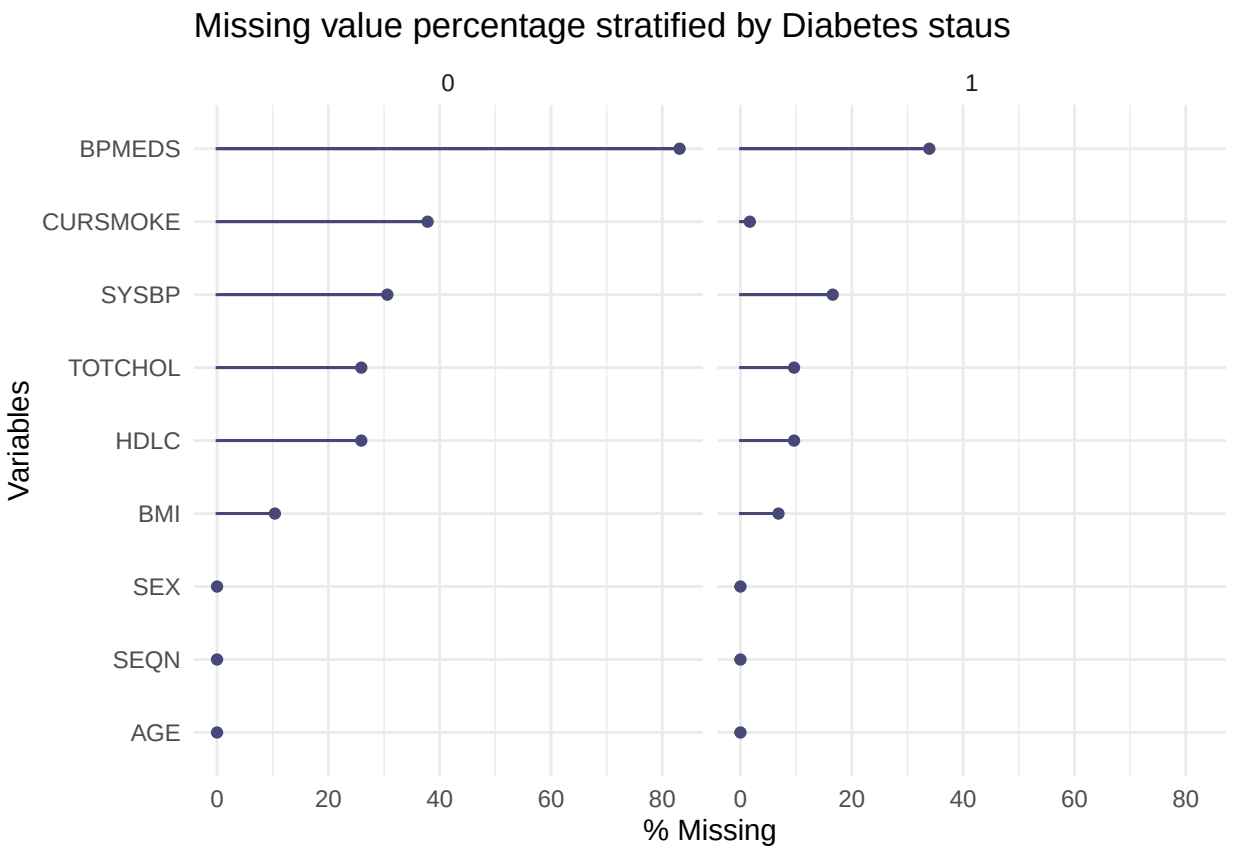
Looking at the comparison table above, we can see that there are significant differences between the covariates of the data between the source and target data. All the p-values that show the significant difference in the variables measured. The table shows that all p-value are significant at level 0.05, except for BMI. This shows that transportability analysis is required to use the framingham model on different distributions of data.

The comparison table also shows that there are a significant number of missing values from the target data. To deal with this we can use multiple imputation methods, however we must assume that the data is missing not at random.

In the tables below we summarize the missing data for each column and observe the missing data patterns stratified by diabetes status(1 = they have diabetes, 0 = no diabetes)

Table 2: Missing Data Proportion for Each Variable

Variable	Observation Missing	Proportion Missing
BPMEDS	7312	79.014480
CURSMOKE	3398	36.719257
SYSBP	2952	31.899719
HDLC	2516	27.188243
TOTCHOL	2516	27.188243
BMI	1249	13.496866
DIABETES	361	3.901016



In the plots above, we can see that there exists significant differences in the number of missing observations

Table 1: **Data Summary stratified by Source and Target Data**

Variable	N	Overall, N = 11,793	Source(1) and Target(0) Data		p-value
			0, N = 9,254	**1**, N = 2,539	
SEX	11,793				<0.001
Female		5,651 / 11,793 (48%)	4,557 / 9,254 (49%)	1,094 / 2,539 (43%)	
Male		6,142 / 11,793 (52%)	4,697 / 9,254 (51%)	1,445 / 2,539 (57%)	
TOTCHOL	9,277	196 (49)	180 (41)	238 (45)	<0.001
Unknown		2,516	2,516	0	
AGE	11,793	40 (25)	34 (26)	60 (8)	<0.001
SYSBP	8,841	127 (22)	121 (20)	140 (23)	<0.001
Unknown		2,952	2,952	0	
CURSMOKE	8,395	1,891 / 8,395 (23%)	1,021 / 5,856 (17%)	870 / 2,539 (34%)	<0.001
Unknown		3,398	3,398	0	
DIABETES	11,432	1,084 / 11,432 (9.5%)	893 / 8,893 (10%)	191 / 2,539 (7.5%)	<0.001
Unknown		361	361	0	
BPMEDS	4,481	2,032 / 4,481 (45%)	1,650 / 1,942 (85%)	382 / 2,539 (15%)	<0.001
Unknown		7,312	7,312	0	
HDLC	9,277	52 (15)	53 (15)	49 (15)	<0.001
Unknown		2,516	2,516	0	
BMI	10,544	26 (7)	27 (8)	26 (4)	0.48
Unknown		1,249	1,249	0	
SYSBP_UT	4,420	74 (70)	137 (46)	116 (52)	<0.001
Unknown		7,373	7,373	0	

between those with diabetes and without diabetes. Because of this missing data pattern we can confirm that the data is missing not at random and continue with multiple imputation.

Evaluation of imputed NHANES data

In the beginning steps of our analysis, we first found variables that are common between the NHANES and the framingham data. These variables were sex, cholesterol levels, age, systolic blood pressure, smoking status, diabetes status, status on whether they take blood pressure or not, high-density lipoproteins(HDL) cholesterol levels and BMI. To evaluate the model on the NHANES data, I followed the steps from these steps from the paper "Transporting a prediction model for use in a new target population":

Let S be an indicator for the population from which data are obtained, with S = 1 for the source population(framingham) and S = 0 for the target population(NHANES). I combined the two population data using the shared common variables as stated above. Next, I split this combined data into a 70-30 training test data split. I used the training set to build a prediction model for the expectation of the outcome(cardiovascular disease) conditional on covariates in the source population. I used the test set to evaluate model performance(MSE) on the target population:

$$\psi_{\hat{\beta}} = E[(Y - g_{\hat{\beta}}(X))|S = 0]$$

.

This is estimated using the following estimator:

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

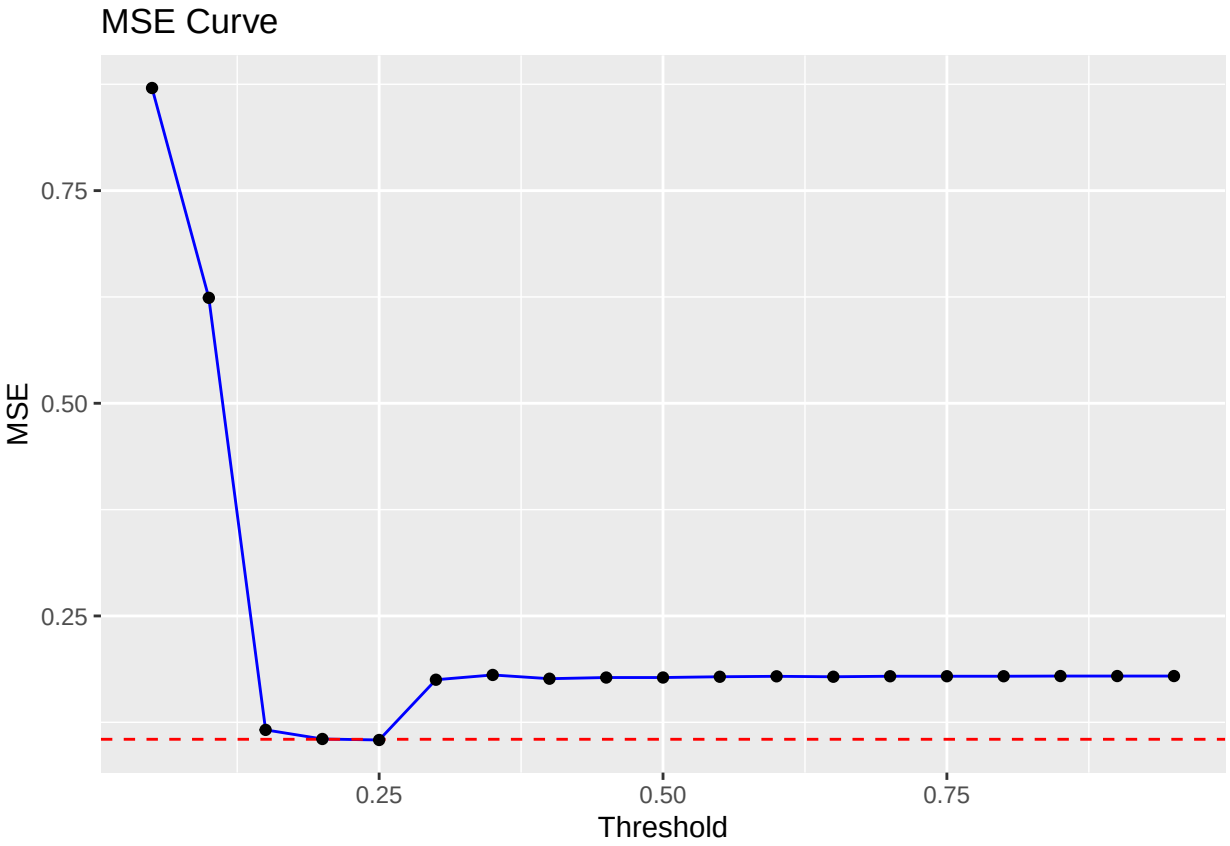
.

where the inverse-odds weights is defined as

$$\hat{o}X_i = \frac{Pr(S = 0|X, D_{test} = 1)}{Pr(S = 1|X, D_{test} = 1)}$$

I calculated the inverse odds weights by training a logistic regression model on the outcome variable(target or population) and then found the specific probabilities for the numerator and denominator using the test set.

The multiple imputation of the NHANES data results in 5 complete datasets. To find the optimal MSE value for the target population, I found the MSE for each data and averaged it across each imputed data. I did this for differing threshold values as well(The threshold value determines whether a predicted outcome is 1 or 0):



The red dashed line represents the MSE calculated when the threshold is found from the ROC(Receiver Operating Characteristic) curve. This value optimizes the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) for different threshold values.

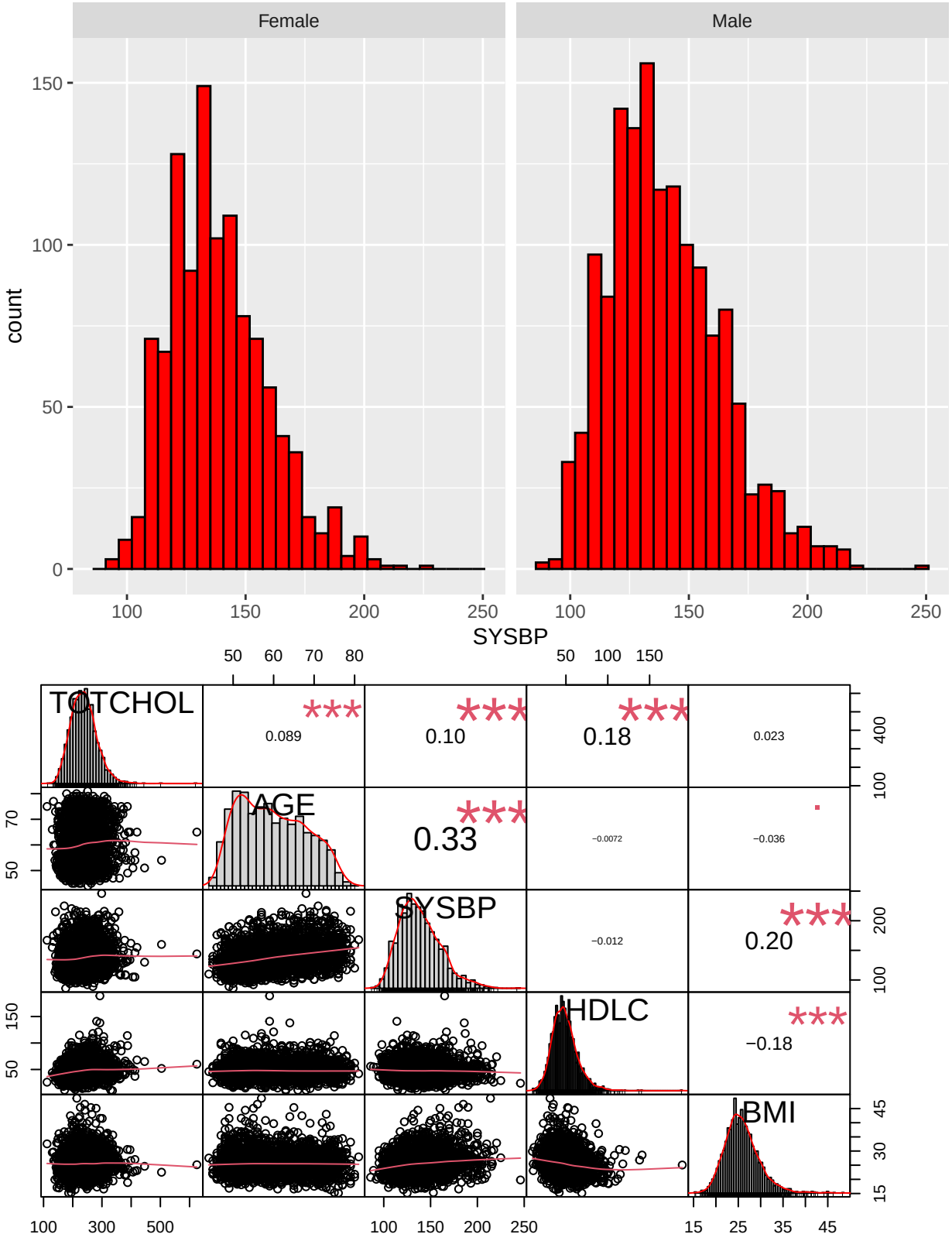
The curve above shows that the model performs well on the target population when the threshold is around 0.25, with an optimal MSE of 0.105.

Simulation:

In the simulation section of the project we will follow the ADEMP framework.

1. **Aims**: The aim of this project is to study compare the evaluation of the simulated data and the actual data over a given set of parameters. Additionally, we want to see how well the simulation performs when we consider associations compared to when we do not.

2. **Data Generation**: To generate data I looked at the framingham data to find associations and distribution relationships between the common variables. I then used this information, in addition to the summary table, and simulated data. I found that most of the variables were not strongly associated with age and sex. An example is shown below:



From the plot above, we see that the distribution of systolic blood pressure is very similar between male and female. Additionally the correlation plot shows that all the continous variables follow a normal distribution approximately and that there exists some significant correlations between the given continous variables. We

use this information to simulate the data.

So, I used the normal distribution(with mean and standard deviation informed by summary statistics table) to simulate age and bmi and the binomial distribution to simulate sex. I found there to be significant linear relationship between systolic blood pressure, age and bmi in the framingham data. Using this information, I set systolic blood pressure to be defined by some linear combinationof age, bmi and some randomly distributed error. I did the same with HDLC(associated with BMI), total cholesterol(associated with age, bmi, sex and systolic blood pressure). To simulate the binary variables(smoking status, blood pressure medicine and diabetes), I calculated the predicted probabilities for each observed value and then used the binomial distribution to convert those probabilities to 1/0.

For reference table 3 in page 8 below compares the summary statistics of the NHANES data and the simulated data:

Comparing the summary statistics of the simulated and actual NHANES data, we can see that they are very similar in terms of the proprotion for categorical variables and sample mean for continous. The standard deviations of the continous variables differ slightly when comparing the NHANES to the simulated set.

3. **Estimand**: The estimand of interest is the $g_{\hat{\beta}}(X))$, where $\hat{\beta}$ is the estimated coefficients from the predictive model g() and X comes from the target population. This is essentially the predicted outcome of the target population based on the covariate X.

4. **Method**: After performing data generation, I followed the steps from the paper as laid out in the methods section to find the mean squared error. I did this over multiple simulations of the data and over differing threshold values. I also compared 2 different data generation process: One where we consider associations between the covariates and another where we don't. Moreover, in the equation of the mean squared error, the term inverse-odds weights is conditioned on X. To study how this will effect the performance of the model simulation, I varied the number of observations per simulation to test to see if the MSE would converge to a number.

5. **Performance Measure**: The Performance Measure of interest is the mean squared error on the simulated(target) population:

$$\psi_{\hat{\beta}} = E[(Y - g_{\hat{\beta}}(X))|S = 0]$$

.

This is estimated using the following estimator:

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

.

where the inverse-odds weights is defined as

$$\hat{o}(X_i) = \frac{Pr(S = 0|X, D_{test} = 1)}{Pr(S = 1|X, D_{test} = 1)}$$

Simulation Results: To test how effective the data generation is, we can simulate data considering associations and simulate data without consider associations and relationships within covariates.

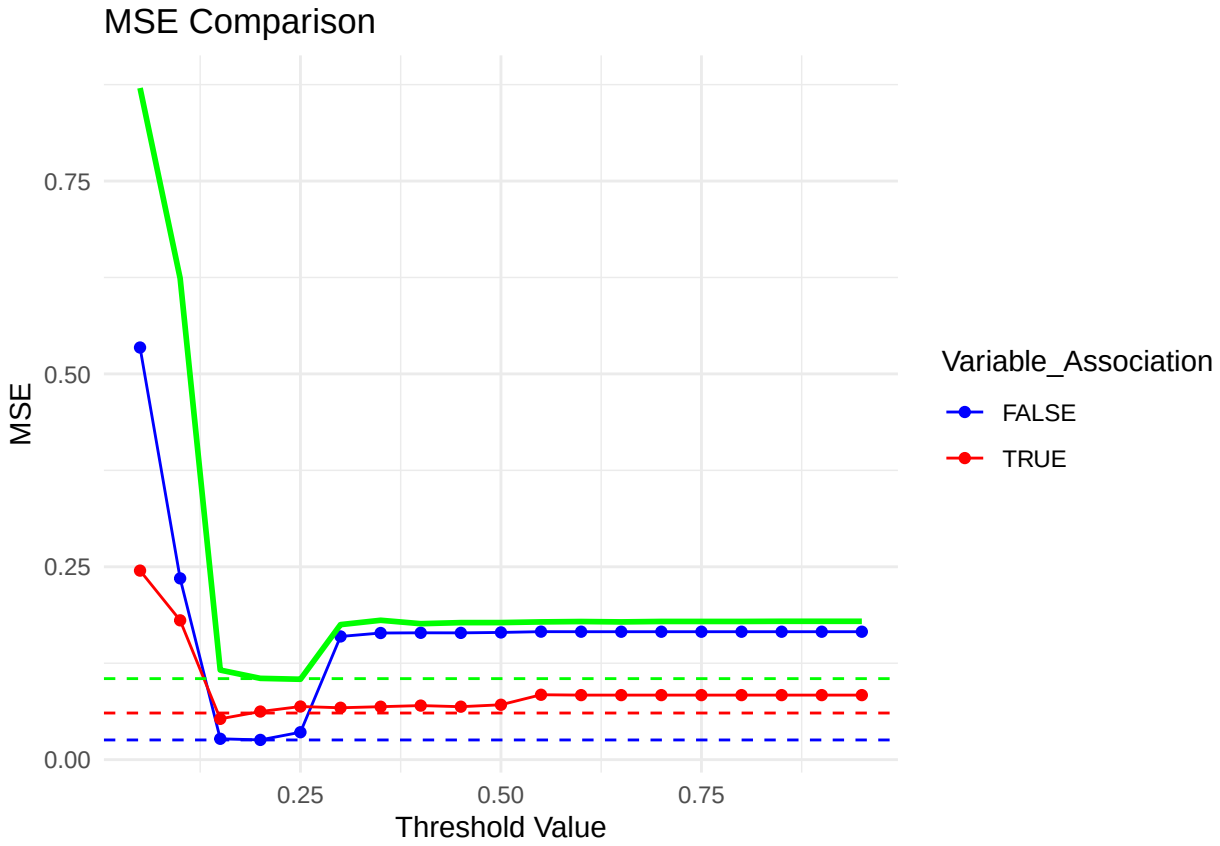
The plot shows how MSE varies by threshold value with and without the consideration of associations between the variables of the data.

Table 3: **Data Summary stratified by simulated and nonsimulated data**

Variable	N	Overall, N = 19,254	Simulated(1) and NHANES(0) Data		p-value
			0, N = 9,254	**1**, N = 10,000	
SYSBP	16,302	121 (29)	121 (20)	121 (33)	0.33
Unknown		2,952	2,952	0	
SEX	19,254				0.30
Female		9,556 / 19,254 (50%)	4,557 / 9,254 (49%)	4,999 / 10,000 (50%)	
Male		9,698 / 19,254 (50%)	4,697 / 9,254 (51%)	5,001 / 10,000 (50%)	
AGE	19,254	34 (26)	34 (26)	34 (26)	0.15
BMI	18,005	27 (8)	27 (8)	27 (8)	<0.001
Unknown		1,249	1,249	0	
HDLC	16,738	53 (16)	53 (15)	53 (16)	0.050
Unknown		2,516	2,516	0	
CURSMOKE	15,856	2,670 / 15,856 (17%)	1,021 / 5,856 (17%)	1,649 / 10,000 (16%)	0.12
Unknown		3,398	3,398	0	
BPMEDS	11,942	10,826 / 11,942 (91%)	1,650 / 1,942 (85%)	9,176 / 10,000 (92%)	<0.001
Unknown		7,312	7,312	0	
TOTCHOL	16,738	180 (43)	180 (41)	180 (44)	<0.001
Unknown		2,516	2,516	0	
DIABETES	18,893	1,759 / 18,893 (9.3%)	893 / 8,893 (10%)	866 / 10,000 (8.7%)	0.001
Unknown		361	361	0	

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

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



Dotted-Solid Red Line: MSE values of simulated data model when we consider associations between variables

Dotted-Solid Blue Line: MSE values of simulated data model when we do not consider associations between variables

Dashed Red Line: MSE value of simulated data model when we consider associations between variables at ROC threshold point

Dashed Blue Line: MSE value of simulated data model when we don't consider associations between variables at ROC threshold point

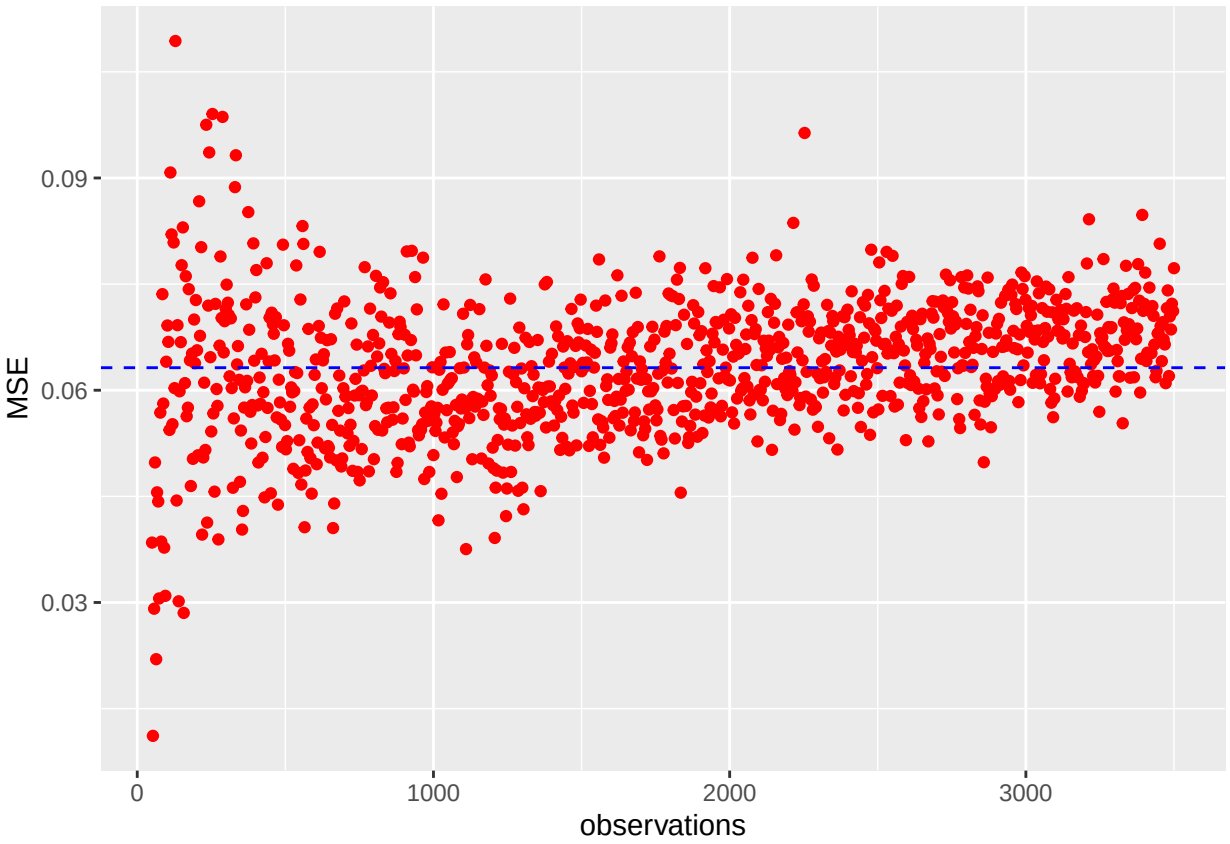
Solid Green Line: MSE values of actual NHANES data model across varying threshold values

Dashed Green Line: MSE values of actual NHANES data model at ROC threshold point.

The plot shows how that when we consider associations of variables, the MSE is lower across all thresholds except between 0.12 and 0.25. The dashed lines represent the MSE value at the ROC curve threshold. It is interesting to see that the the ROC curve threshold MSE is lower when we do not consider variable association compared to when we do.

The solid green line represents the MSE values of the NHANES data. We can see that when we perform evaluation on the model, the model performs better on the simulated data compared to the actual NHANES data.

Next we consider how the number of observations in the simulated data can effect the mean squared error.




```
## NULL
```

The plot shows that as the number of observations increase, the MSE converges towards the mean of the simulated MSE's(represented by the blue dashed line). This may be because as the number of observations increase, the model variation decreases. Additionally we expect reduced sampling error and improved model generalization due to this factor.

Discussion and Conclusion

In conclusion, in this project we have first evaluated the model on the target population using the MSE as the main performance metric. We found that the model evaluates well on the framingham data achieving an MSE score of around 0.105. We then studied the associations and distributions of the common variables in the framingham data and used this to simulate a second dataset in the hope of mirroring the NHANES data. To inform the parameters of the distribution we used a summary table from the NHANES dataset that provided us with useful information such as the mean, standard deviation and proportions of the covariates. Using this datagenerating process, we simulated data considering both associations and no associations and also under different threshold values. We found that the model evaluates well on the NHANES data achieving an optimal MSE score of around 0.105, however it performs better on the simulated data.

Some limitations of my study is that I only considered one metric to evaluate the performance of the model evaluation on the target population. One metric I could have used is the AUC performance metric in transportability analysis(from the paper "Estimating the area under the ROC curve when transporting a prediction model to a target population"), however it has been proven difficult to implement.

Code Appendix

```
options(warn=-1)

library(riskCommunicator)
library(tidyverse)
library(tableone)
library(dplyr)
library(nhanesA)
library(mice)
library(naniar)
library(pROC)
library(PerformanceAnalytics)
library(dplyr)
library(ggplot2)
library(HDSinRdata)
library(tidyverse)
library(egg)
library(tableone)
library(mice)
library(naniar)
library(gt)
library(gtsummary)
library(kableExtra)
library(lme4)
library(reshape2)
library(StatisticalModels)
library(glmmLasso)
library(pROC)
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
# This paper is available (cvd_risk_profile.pdf) on Canvas.

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

CreateTableOne(data=framingham_df, strata = c("SEX"))

# 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)) %>%
```

```
  dplyr::select(-c(TIMECVD))
dim(framingham_df)

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

#split into training and test sets
sample.men <- sample(c(TRUE, FALSE), nrow(framingham_df_men), replace=TRUE, prob=c(0.7,0.3))
train.men  <- framingham_df_men[sample.men, ]
test.men   <- framingham_df_men[!sample.men, ]

sample.women <- sample(c(TRUE, FALSE), nrow(framingham_df_women), replace=TRUE, prob=c(0.7,0.3))
train.women  <- framingham_df_women[sample.women, ]
test.women   <- framingham_df_women[!sample.women, ]
# Fit models with log transforms for all continuous variables
mod_men <- glm(CVD~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                 log(SYSBP_T+1)+CURSMOKE+DIABETES,
      data= 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= train.women, family= "binomial")


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

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

```
  mutate(DIABETES = case_when(DIQ010 == 1 ~ 1,
                              DIQ010 %in% c(2,3) ~ 0,
                              TRUE ~ NA)) %>%
  dplyr::select(SEQN, DIABETES)

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

CreateTableOne(data = df_2017, strata = c("SEX"))

#multiple imputation
varMissingProp = miss_var_summary(df_2017)
varMissingProp %>%
  filter(n_miss > 0) %>%
  kableExtra::kbl(caption = 'Missing Data Proportion for Each Variable'
                  , booktabs = T
                  , escape = T
                  , align = 'c'
                  , col.names = c('Variable','Observation Missing','Proportion
                                   Missing')) %>%
  kableExtra::kable_classic(full_width = F
                            , html_font = 'Cambria'
                            , latex_options = 'HOLD_position')
#multiple imputation
imp_2107 <- mice(df_2017, 5, pri=F)
comp_imp_2017 = mice::complete(imp_2107,1)
framingham_df["S"] = 1
comp_imp_2017["S"] = 0


comp_imp_2017$SYSBP_UT <- ifelse(comp_imp_2017$BPMEDS == 0,
                                 comp_imp_2017$SYSBP, 0)
comp_imp_2017$SYSBP_T <- ifelse(comp_imp_2017$BPMEDS == 1,
                                comp_imp_2017$SYSBP, 0)
comp_imp_2017$CVD = NA
df_2017.2 = df_2017
df_2017.2["S"] = 0


df_2017.2$SYSBP_UT <- ifelse(df_2017.2$BPMEDS == 0,
                                 df_2017.2$SYSBP, 0)
df_2017.2$SYSBP_T <- ifelse(df_2017.2$BPMEDS == 1,
                                df_2017.2$SYSBP, 0)
df_2017.2$CVD = NA

common_columns <- intersect(names(framingham_df), names(df_2017.2))
```

```r
print(common_columns)


combined_df = rbind(framingham_df[,common_columns], df_2017.2[,common_columns])
comb.df = combined_df
comb.df$SEX <- factor(comb.df$SEX, levels = c(1, 2), labels = c("Female", "Male"))

summary_strat.comb = comb.df[,-1] %>%
  tbl_summary( by = S,   statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} / {N} ({p}%)"
    )) %>%
  add_p(pvalue_fun = ~ style_pvalue(.x, digits = 2)) %>%
  add_overall() %>%
  add_n() %>%
  modify_header(label ~ "**Variable**") %>%
  modify_spanning_header(c("stat_1", "stat_2") ~ "**Source(1) and Target(0) Data**") %>%
  modify_caption("**Data Summary stratified by Source and Target Data **") %>%
  bold_labels() %>%
  as_kable_extra(booktabs = TRUE) %>%
  kableExtra::kable_styling(latex_options = "scale_down")%>%
   column_spec(1,width = "1in") %>%
  column_spec(2,width = "0.4in") %>%
  column_spec(3,width = "0.4in") %>%
  column_spec(4,width = "0.4in")
summary_strat.comb
summary_strat.comb

varMissingProp %>%
  filter(n_miss > 0) %>%
  kableExtra::kbl(caption = 'Missing Data Proportion for Each Variable'
                  , booktabs = T
                  , escape = T
                  , align = 'c'
                  , col.names = c('Variable','Observation Missing','Proportion
                                  Missing')) %>%
  kableExtra::kable_classic(full_width = F
                            , html_font = 'Cambria'
                            , latex_options = 'HOLD_position')
df_2017[!is.na(df_2017$DIABETES), ] %>%
  gg_miss_var(show_pct = TRUE, facet = DIABETES)  + ggtitle("Missing value percentage stratified by Dia
brier_score = function(data, threshold_logic, thresh_val){
comp_imp_2017 = data
framingham_df["S"] = 1
comp_imp_2017["S"] = 0

comp_imp_2017$SYSBP_UT <- ifelse(comp_imp_2017$BPMEDS == 0,
                                 comp_imp_2017$SYSBP, 0)
comp_imp_2017$SYSBP_T <- ifelse(comp_imp_2017$BPMEDS == 1,
                                comp_imp_2017$SYSBP, 0)
comp_imp_2017$CVD = NA
common_columns <- intersect(names(framingham_df), names(comp_imp_2017))
#print(common_columns)
```

```
combined_df = rbind(framingham_df[,common_columns], comp_imp_2017[,common_columns])
set.seed(1)
sample <- sample(c(TRUE, FALSE), nrow(combined_df), replace=TRUE, prob=c(0.7,0.3))
train <- combined_df[sample, ]
test   <- combined_df[!sample,]

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

pred =   predict(mod, newdata = test[test$S ==1,], type = "response")
roc_curve <- roc(response = test[test$S ==1,]$CVD, predictor = pred)
#plot(roc_curve, print.auc=TRUE, print.thres = TRUE)
if(threshold_logic == TRUE){
  thresh = as.numeric(coords(roc_curve, "best", ret = "threshold"))
} else {
  thresh = thresh_val
}
pred_ys <- ifelse(pred > thresh, 1, 0)
weight.io = glm(S~., data=train[,-1] , family= "binomial") #Warning: glm.fit: algorithm did not converg
predicted_probs <- predict(weight.io, newdata = test[test$S ==1,], type = "response")

io_weights = (1-predicted_probs)/(predicted_probs)

num = sum(io_weights*(test[test$S ==1,]$CVD - pred_ys)^2)
# Plot a kernel density plot of the probability ratio
den = sum(test$S ==0)
mse = num/den
return(mse)
}

iter = seq(0.05, 0.95, by = 0.05)
bs.mat = matrix(NA, nrow = 5,ncol = length(iter))
colnames(bs.mat) = iter
for(i in 1:5){
  df_brier = mice::complete(imp_2107,i)
  bs.vec = c()
  for(k in iter){
    bs = brier_score(df_brier, FALSE, thresh_val=k) #brier score
    bs.vec = c(bs.vec, bs) #create vector of brier scores
  }
   bs.mat[i,] = bs.vec
}

bs.mean.iter = colMeans(bs.mat)
plot(iter, bs.mean.iter)
#brier_score(mice::complete(imp_2107,1), FALSE, thresh_val=0.5)

#mean using the ROC threshold value:
bs.vec = c()
for(i in 1:5){
  df_brier = mice::complete(imp_2107,i)
```

```
  bs = brier_score(df_brier, TRUE, thresh_val=k) #brier score
  bs.vec = c(bs.vec, bs) #create vector of brier scores
  }
bs.mean = mean(bs.vec)
#plot(iter, bs.mean.iter)
ggplot(data.frame(iter, bs.mean.iter), aes(x = iter, y = bs.mean.iter)) +
  geom_line(color = "blue") +
  geom_hline(yintercept = bs.mean, linetype = "dashed", color = "red") +
  geom_point() +
  labs(title = "MSE Curve",
       x = "Threshold",
       y = "MSE")

options(warn=-1)

ggplot(framingham_df, aes(x = SYSBP)) +
  geom_histogram(fill = "red", colour = "black",bins=30 ) +
  facet_grid(. ~ factor(SEX, labels = c("Female", "Male")))   # Use labels argument to change facet lab


chart.Correlation(framingham_df[,common_columns[c(3:5,9:10)]], histogram = TRUE, method = "pearson")
summary_strat = df_2017[,-1] %>%
  tbl_summary(statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} / {N} ({p}%)"
    ))


simulate_data = function(m,seed=1, logic_param){
if(logic_param == TRUE){
set.seed(seed)
AGE = rnorm(m, mean = 34, sd = 26)
BMI = rnorm(m, mean = 27, sd = 8)
# Simulate sex using binomial distribution
SEX = rbinom(m, size = 1, prob = 0.49)
SYSBP = 0.92477*AGE + 1.22478*BMI + rnorm(m, 0, 20) #sys blood pressure
int.SYSBP = 121 - mean(SYSBP)
SYSBP = SYSBP+int.SYSBP
HDLC = 0.7118*BMI+ rnorm(m, 0, 15)
int.HDLC = 53 - mean(HDLC)
HDLC = HDLC+int.HDLC
CURSMOKE = rbinom(m, size = 1, prob = 0.17)

TOTCHOL = 0.32725 *AGE + 0.58230*BMI + 16.09506*SEX+0.13679*SYSBP+  0.40140 *HDLC +  rnorm(m, 0, 41)
int.TOTCHOL = 180- mean(TOTCHOL)
TOTCHOL = TOTCHOL+int.TOTCHOL
BP_MED = framingham_df %>% dplyr::select(c(SYSBP, BPMEDS))
breaks <- c(80, 130, 150, 180, 210,240,270 , Inf)

# Create a new variable for the bins
BP_MED$ContinuousVariableBin <- cut(BP_MED$SYSBP, breaks = breaks)

# Create a table counting occurrences of 1's in the binary column by bin
```

```r
table_result <- table(BP_MED$ContinuousVariableBin, BP_MED$BPMEDS)
table_result = as.data.frame.matrix(table_result)
table_result$P <- table_result$`1` / rowSums(table_result[, c("0", "1")])
new_probabilities <- approx(
  x = c(80, 130, 150, 180, 210, 240, Inf),
  y = table_result$P,
  xout = SYSBP,
  rule = 2
)$y
BPMEDS<- rbinom(length(SYSBP), size = 1, prob = new_probabilities^(1/12))

b_mod = glm(BPMEDS~SYSBP, family = "binomial"(logit), data = framingham_df)
predicted_probabilities <- predict(b_mod, newdata = as.data.frame(SYSBP), type = "response")

# Convert probabilities to binary smoking status
BPMEDS <- rbinom(m, size = 1, prob = predicted_probabilities^0.03)
#ggplot(framingham_df, aes(x = BMI)) +
#  geom_histogram(fill = "white", colour = "black") +
#  facet_grid(SEX ~ .)



d_mod = glm(DIABETES~BMI+SYSBP, family = "binomial"(logit), data = framingham_df)
predicted_probabilities <- predict(d_mod, newdata = as.data.frame(BMI, SYSBP), type = "response")

# Convert probabilities to binary smoking status
DIABETES <- rbinom(m, size = 1, prob = predicted_probabilities^.9)
#mean(DIABETES)
# Convert to factor #Age and BMI are same between men and women

  }else{
AGE = rnorm(m, mean = 34, sd = 26)
BMI = rnorm(m, mean = 27, sd = 8)
# Simulate sex using binomial distribution
SEX = rbinom(m, size = 1, prob = 0.49)
SYSBP = rnorm(m, 121, 20) #sys blood pressure
HDLC = rnorm(m, 53, 15)
CURSMOKE = rbinom(m, size = 1, prob = 0.17)
TOTCHOL = rnorm(m, 180,41)
DIABETES = rbinom(m, size = 1, prob = 0.1)
SYSBP = rnorm(m, mean = 121, sd = 20)
BPMEDS = rbinom(m, size = 1, prob = 0.85)

  }
sim_data = data.frame(SYSBP, SEX, AGE, BMI, HDLC, CURSMOKE, BPMEDS, TOTCHOL, DIABETES)
return(sim_data)
}

sim_data.T = simulate_data(m=2000,seed=1, logic_param = TRUE)
sim_data.F = simulate_data(m=2000,seed=1, logic_param = FALSE)
iter = seq(0.05, 0.95, by = 0.05)
bs.vec.T =c()
bs.vec.F =c()
```

```r
for(k in iter){
    bs.T = brier_score(sim_data.T, FALSE, thresh_val=k) #brier score
    bs.F = brier_score(sim_data.F, FALSE, thresh_val=k) #brier score
    bs.vec.T = c(bs.vec.T, bs.T) #create vector of brier scores
    bs.vec.F = c(bs.vec.F, bs.F) #create vector of brier scores
}

mse_data <- data.frame(Iteration = iter,
                       MSE = c(bs.vec.T, bs.vec.F),
                       Variable_Association = rep(c("TRUE", "FALSE"), each = length(iter)))



# Plot using ggplot2
bs.roc.T = brier_score(sim_data.T, TRUE, thresh_val=k) #brier score
bs.roc.F = brier_score(sim_data.F, TRUE, thresh_val=k) #brier score


#brier_score(mice::complete(imp_2107,1), FALSE, thresh_val=0.5)




sim_data.T.3 = simulate_data(m=10000,seed=1, logic_param = TRUE)
sim_data.T.3$SEX = factor(sim_data.T.3$SEX, levels = c(0, 1), labels = c("Female", "Male"))
sim_data.T.3["Simulated"] = 1
df_2017.3 =  df_2017
df_2017.3["Simulated"] = 0
df_2017.3$SEX <- factor(df_2017.3$SEX, levels = c(1, 2), labels = c("Female", "Male"))


common_columns.2 <- intersect(names(sim_data.T.3), names(df_2017.3))
print(common_columns.2)

comp_df = rbind(df_2017.3[,common_columns.2], sim_data.T.3[,common_columns.2])

summary_strat.comp = comp_df %>%
  tbl_summary( by = Simulated,   statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} / {N} ({p}%)"
    )) %>%
  add_p(pvalue_fun = ~ style_pvalue(.x, digits = 2)) %>%
  add_overall() %>%
  add_n() %>%
  modify_header(label ~ "**Variable**") %>%
  modify_spanning_header(c("stat_1", "stat_2") ~ "**Simulated(1) and NHANES(0) Data**") %>%
  modify_caption("**Data Summary stratified by simulated and nonsimulated data **") %>%
  bold_labels() %>%
  as_kable_extra(booktabs = TRUE) %>%
  kableExtra::kable_styling(latex_options = "scale_down")%>%
   column_spec(1,width = "1in") %>%
  column_spec(2,width = "0.4in") %>%
  column_spec(3,width = "0.4in") %>%
```

```
  column_spec(4,width = "0.4in")
summary_strat.comp
summary_strat.comp
ggplot(mse_data, aes(x = Iteration, y = MSE, color = Variable_Association, linetype = Variable_Associat
  geom_line() +
  geom_point() +
  geom_hline(yintercept = c(bs.roc.T, bs.roc.F, bs.mean), linetype = "dashed", color = c("red", "blue",
  labs(title = "MSE Comparison",
       x = "Iteration",
       y = "MSE") +
  scale_color_manual(values = c("blue", "red")) +
  scale_linetype_manual(values = c("solid", "solid")) +
  xlab("Threshold Value")+
  theme_minimal()+
  geom_line(data = data.frame(iter, bs.mean.iter), aes(x = iter, y = bs.mean.iter), color = "green", si



n_sim = 1000
br.score.vec = c()
m_vec <- round(seq(from = 50, to = 3500, length.out = n_sim))

for(i in 1:n_sim){
  ran.seed = sample(seq(from = -1000, to = 1000), 1)
  sim_data = simulate_data(m = m_vec[i],seed=ran.seed, logic_param= TRUE)
  br = brier_score(sim_data, TRUE, thresh_val=NA)
  br.score.vec = c(br.score.vec, br)
  print(i)
}
br_data <- data.frame(observations = m_vec,
                      MSE = br.score.vec)

ggplot(br_data, aes(x = observations, y = MSE)) +
  geom_point(color = "red") +
  geom_hline(yintercept = mean(br.score.vec),linetype = "dashed", color = "blue")
  labs(title = "MSE vs. Number of observations",
       x = "observations",
       y = "MSE") +
  scale_color_manual(values = c("red")) +
  scale_linetype_manual(values = c("solid")) +
  theme_minimal()
```