Practical Data Analysis Project 3

Tabib Chowdhury

2023-12-16

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 0.075 for males and 0.042 for femlaes. Comparatively, the MSE on the framingham data is lower at 0.157 for males and 0.155 for females. 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. Table 1 below compares the common variables between the two dataset.

Looking at the comparison table 1, 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.

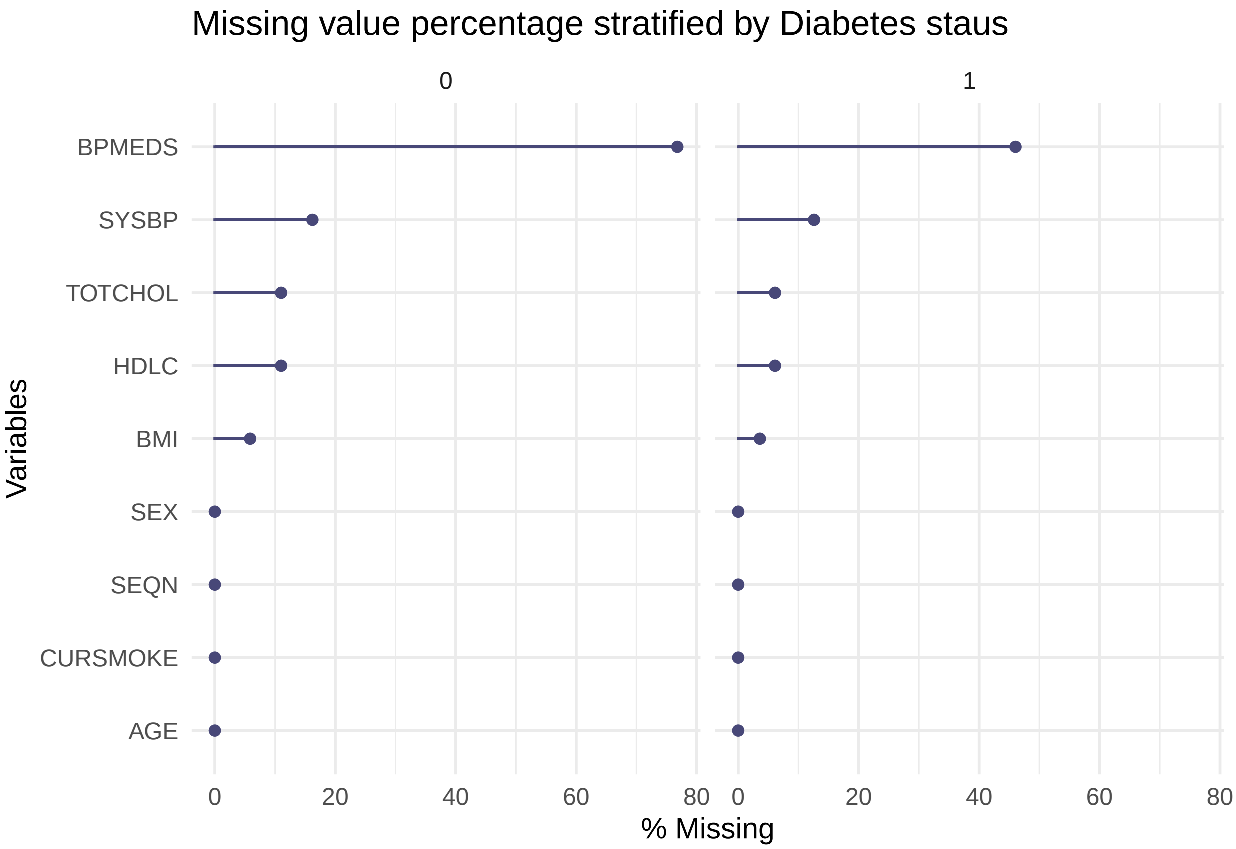
In the tables below we summarize the missing data for each column and observe the missing data patterns stratified by diabetes status(1 = they have diabetes, 0 = no diabetes). We decided to stratify by diabetes since it is related to the outcome variable of CVD.

Table 2: Missing Data Proportion for Each Variable

Variable	Observation Missing	Proportion Missing
BPMEDS	1949	73.5749339
SYSBP	419	15.8172895
HDLC	278	10.4945262
TOTCHOL	278	10.4945262
BMI	149	5.6247641
DIABETES	1	0.0377501

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

			Source(1) and Target(0) Data		
Variable	**N**	**Overall**, N = 5,188	**0**, N = 2,649	**1**, N = 2,539	**p-value**
SEX	5,188				0.007
Female		2,335 / 5,188 (45%)	1,241 / 2,649 (47%)	1,094 / 2,539 (43%)	
Male		2,853 / 5,188 (55%)	1,408 / 2,649 (53%)	1,445 / 2,539 (57%)	
TOTCHOL	4,910	217 (47)	195 (39)	238 (45)	<0.001
Unknown		278	278	0	
AGE	5,188	53 (11)	46 (9)	60 (8)	<0.001
SYSBP	4,769	132 (22)	124 (18)	140 (23)	<0.001
Unknown		419	419	0	
CURSMOKE	5,188	1,420 / 5,188 (27%)	550 / 2,649 (21%)	870 / 2,539 (34%)	<0.001
DIABETES	5,187	469 / 5,187 (9.0%)	278 / 2,648 (10%)	191 / 2,539 (7.5%)	<0.001
Unknown		1	1	0	
BPMEDS	3,239	912 / 3,239 (28%)	530 / 700 (76%)	382 / 2,539 (15%)	<0.001
Unknown		1,949	1,949	0	
HDLC	4,910	51 (16)	53 (16)	49 (15)	<0.001
Unknown		278	278	0	
BMI	5,039	28 (7)	31 (8)	26 (4)	<0.001
Unknown		149	149	0	
SYSBP_UT	3,207	98 (64)	29 (58)	116 (52)	<0.001
Unknown		1,981	1,981	0	
SYSBP_T	3,152	38 (65)	96 (61)	24 (57)	<0.001
Unknown		2,036	2,036	0	



In the plots above, we can see that there exists significant differences in the number of missing observations between those with diabetes and without diabetes. Because of this missing data pattern we can confirm that the data is missing 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 (Steingrimsson et al., 2023).

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.

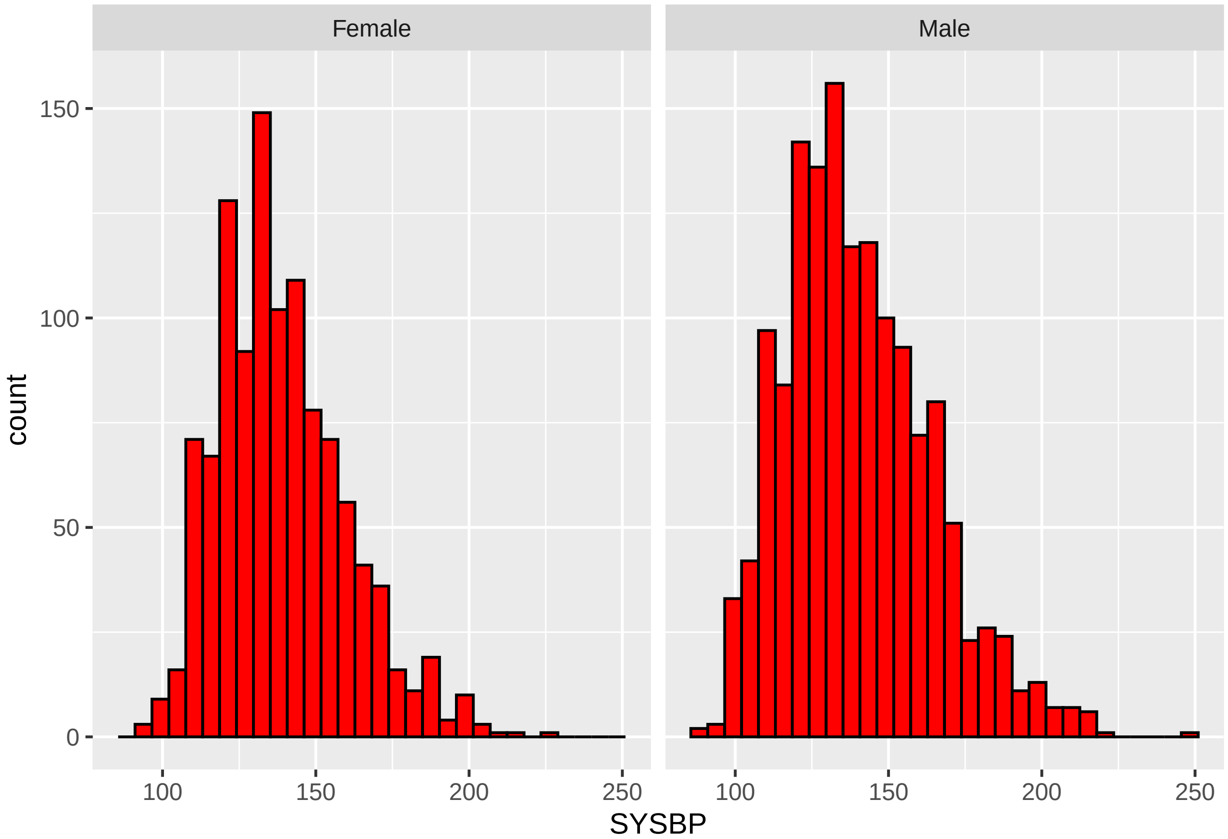
The table above compares MSE(brier scores) of females and males across the source and target population. We find that the transportability analysis performs well on the female gender compared to males. Additionally we see that the target population performs better than the source as the MSE is almost doubled in the source population. 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.

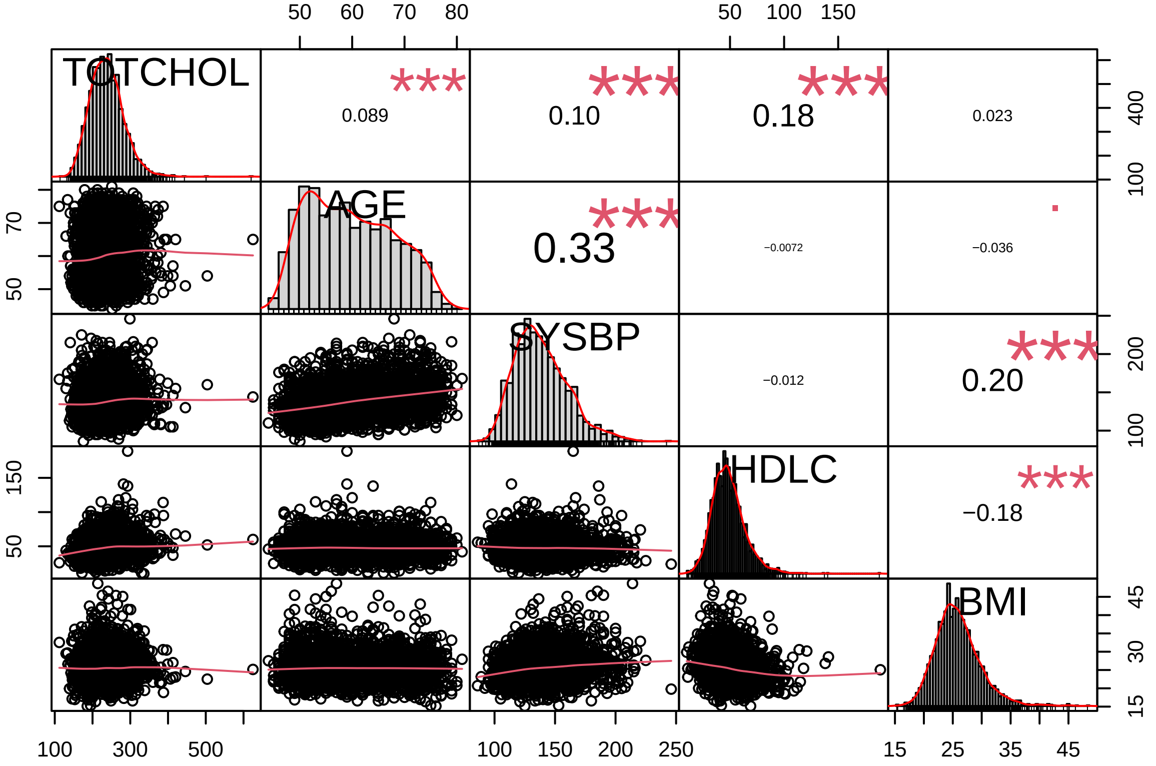
Simulation:

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. We simulate over different levels of associations comparing data that has strong associations vs weak associations between the variables

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 multivariate 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 defined the covaraince matrix such that there is some relationships and associations between these variables. 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 used the multivariate binomial distribution with an attached binomial correlation matrix to capture associations.

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 probability 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 association levels. More specifically, the level associations refers to the how close the correlation coefficients are to 1. I also compared 2 different data generation process: One where we consider associations between the covariates and another where we don't. I completed these procedures for both the female and male genders to compare how transportability analysis compares across gender.

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

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

I also calculate standard errors of the MSE across simulation to see how variable this performance measure is.

Simulation Results: To test how effective the data generation is, we can simulate data considering associations levels.

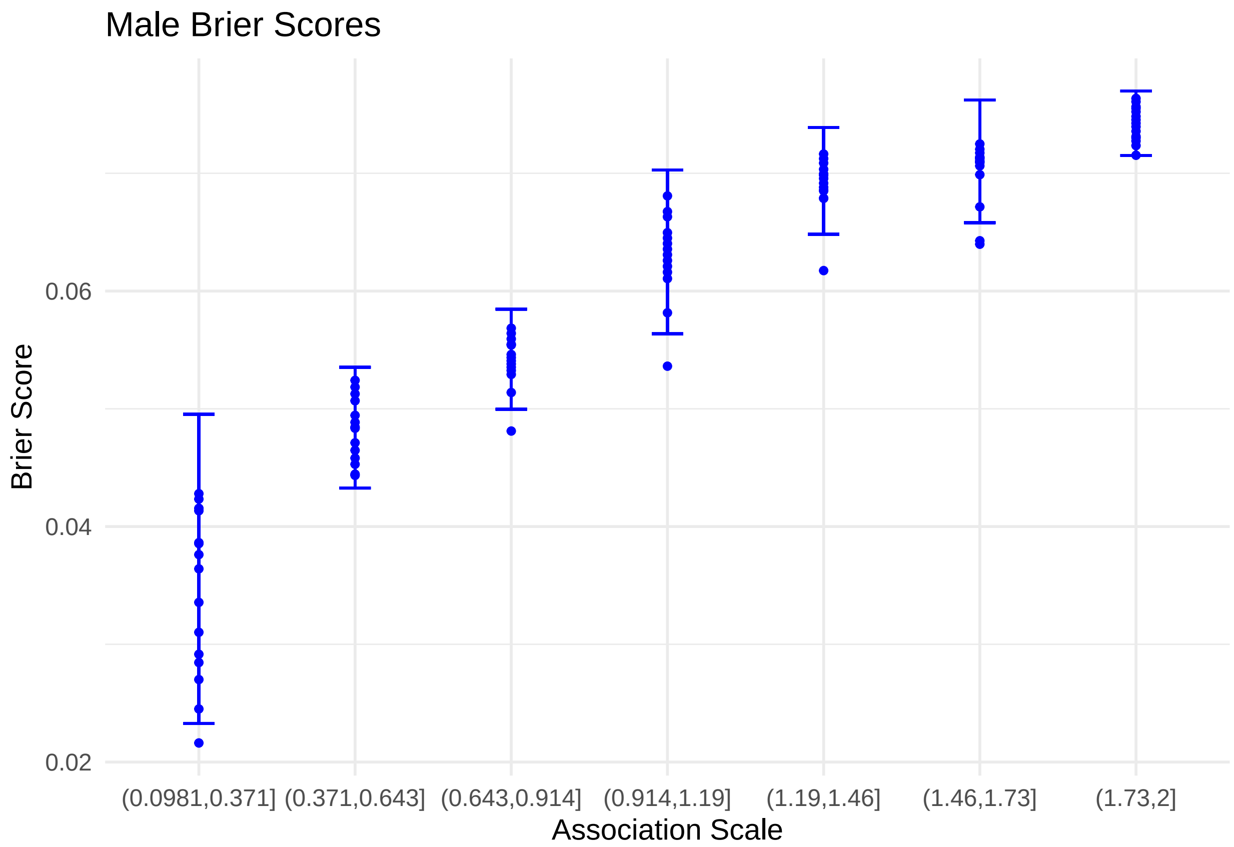
The plot shows how MSE varies by the levels of association across males and females.

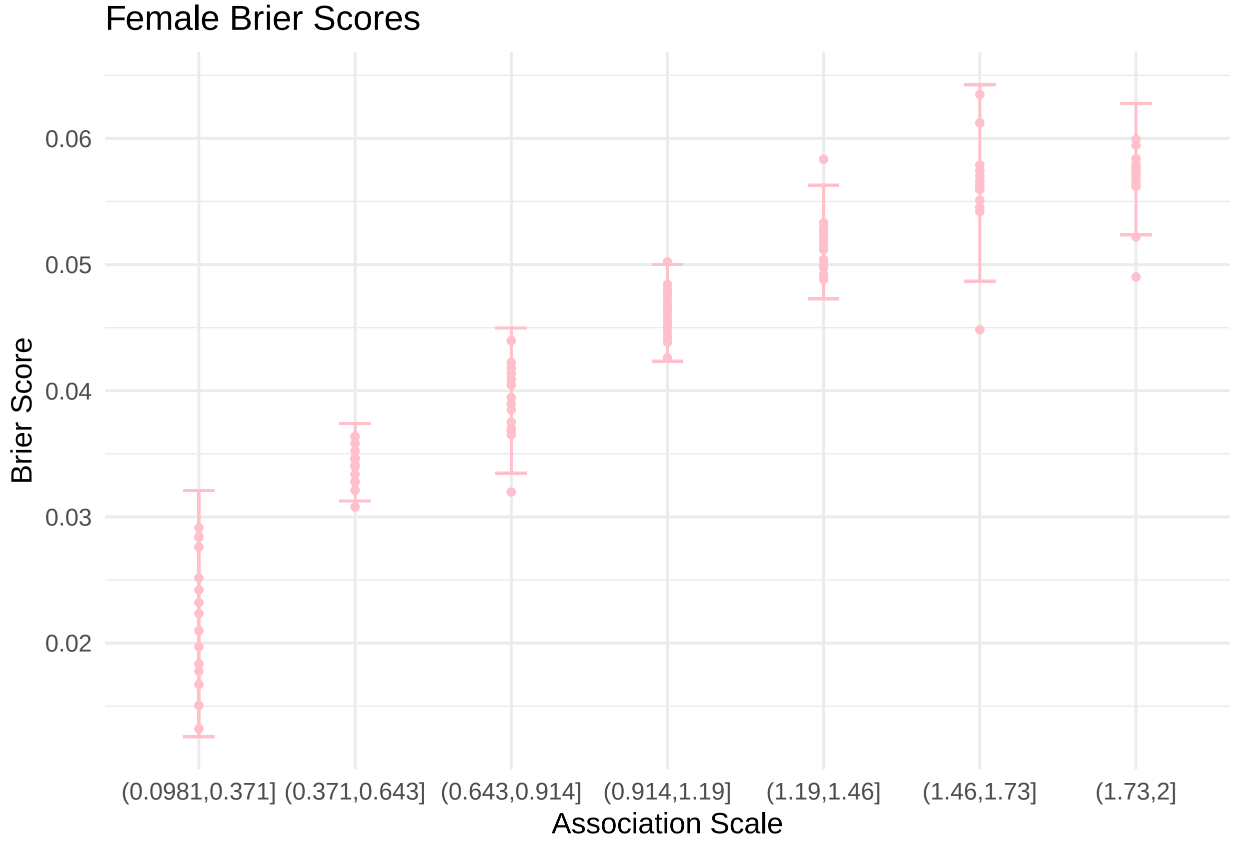
Table 3: **Data Summary stratified by simulated and nonsimulated data**

			Simulated(1) and NHANES(0) Data		
Variable	**N**	**Overall**, N = 3,649	**0**, N = 2,649	**1**, N = 1,000	**p-value**
TOTCHOL	3,371	195 (48)	195 (39)	194 (66)	0.87
Unknown		278	278	0	
AGE	3,649	46 (10)	46 (9)	47 (12)	0.90
SYSBP	3,230	123 (22)	124 (18)	122 (30)	0.11
Unknown		419	419	0	
HDLC	3,371	53 (18)	53 (16)	53 (22)	0.11
Unknown		278	278	0	
BMI	3,500	30 (7)	31 (8)	30 (6)	0.002
Unknown		149	149	0	
CURSMOKE	3,649	743 / 3,649 (20%)	550 / 2,649 (21%)	193 / 1,000 (19%)	0.33
BPMEDS	1,700	1,284 / 1,700 (76%)	530 / 700 (76%)	754 / 1,000 (75%)	0.88
Unknown		1,949	1,949	0	
DIABETES	3,648	396 / 3,648 (11%)	278 / 2,648 (10%)	118 / 1,000 (12%)	0.26
Unknown		1	1	0	
SEX	3,649				0.079
Female		1,742 / 3,649 (48%)	1,241 / 2,649 (47%)	501 / 1,000 (50%)	
Male		1,907 / 3,649 (52%)	1,408 / 2,649 (53%)	499 / 1,000 (50%)	

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

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





In the plots above the general observation when comparing the simulated datasets across different levels of associations is that as the associations get stronger the brier score increases. However we see that the standard error condfidence intervals decrease.

The table of standard errors and mean brier scores across different simulations of associations is shown below.

Association_interval	Male_Brier_Scores	Female_Brier_Scores	Male_SE	Female_SE
(0.0981,0.371]	0.0343041	0.0220207	0.0066981	0.0049776
(0.371,0.643]	0.0481992	0.0341101	0.0026176	0.0015672
(0.643,0.914]	0.0540084	0.0391049	0.0021682	0.0029387
(0.914,1.19]	0.0628849	0.0461928	0.0035483	0.0019599
(1.19,1.46]	0.0689985	0.0518122	0.0023139	0.0022939
(1.46,1.73]	0.0699307	0.0562408	0.0026584	0.0039736
(1.73,2]	0.0741813	0.0567518	0.0013973	0.0026575

We find that the table gives us the same result as the plots. As the association levels increase, the brier scores increase and the standard errors decrease.

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 and comparatively poorer on the framingham data. We found that the model evaluates well on the NHANES data achieving an MSE score of around 0.075 for males and 0.042 for femlaes. Comparatively, the MSE on the framingham data islower at 0.157 for males and 0.155 for females. In addition it was found that the model performs better on females compared to males. 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. Model performance and evaluations were then visualized with the tables and plots above. Overall the simulated data resulted in similar MSE scores however we found that as associations got stronger, the MSE increased and standard error decreased.

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.

References

1) Steingrimsson JA, Gatsonis C, Li B, Dahabreh IJ. Transporting a Prediction Model for Use in a New Target Population. Am J Epidemiol. 2023 Feb 1;192(2):296-304. doi: 10.1093/aje/kwac128. PMID: 35872598.

Code Appendix

```
options(warn=-1)
library(bindata)
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)
```

```r
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)

#heart attack
hrt_str_2017 <- nhanes("MCQ_J") %>%
  mutate(HRT_ATTCK = case_when(MCQ160E == 1 ~ 1,
                              MCQ160E %in% c(2) ~ 0,
                              TRUE ~ NA))%>%
    mutate(STROKE = case_when(MCQ160F == 1 ~ 1,
                              MCQ160F %in% 2 ~ 0,
                              TRUE ~ NA))%>%
  dplyr::select(SEQN, HRT_ATTCK,STROKE)



#stroke

# 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") %>%
  full_join(hrt_str_2017, by = "SEQN")

df_2017 = subset(df_2017, HRT_ATTCK == 0 & STROKE == 0)
df_2017 = subset(df_2017, AGE<62 & AGE>30)
df_2017 = subset(df_2017, select = -c(HRT_ATTCK, STROKE))
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))
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
```

```r
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$DIABETES), ] %>%
  gg_miss_var(show_pct = TRUE, facet = DIABETES)  + ggtitle("Missing value percentage stratified by Dia
#brier_score = function(data, threshold_logic, thresh_val, sex){
#comp_imp_2017 = data
brier_score = function(data, SEX){
  comp_imp_2017 = data
  sex = SEX
  fram = framingham_df
  fram["S"] = 1
  comp_imp_2017["S"] = 0
  fram=fram[fram$SEX == sex,]
  comp_imp_2017 = comp_imp_2017[comp_imp_2017$SEX == sex,]
  comp_imp_2017$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(fram), names(comp_imp_2017))


  combined_df = framingham_df[,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)+factor(CURSMOKE)+factor(DIABETES),
        data=train, family= "binomial")

  pred =   predict(mod, newdata = test, type = "response")


  common_columns <- intersect(names(test), names(comp_imp_2017))

  test_comb = rbind(test[,common_columns], comp_imp_2017[,common_columns])
  test_comb = subset(test_comb, AGE != 0)

  weight.io = glm(factor(S)~log(HDLC)+log(TOTCHOL)+log(AGE)+log(SYSBP_UT+1)+
                   log(SYSBP_T+1)+factor(CURSMOKE)+factor(DIABETES), data=test_comb, family= "binomial"
  predicted_probs <- predict(weight.io, newdata = test_comb[test_comb$S ==1,], type = "response")

  io_weights = (1-predicted_probs)/(predicted_probs)

  num = sum(io_weights*(test_comb[test_comb$S ==1,]$CVD - pred)^2)
  den = sum(test_comb$S ==0)
  mse = num/den
  return(mse)
}

bs.nhanes = data.frame(male = numeric(5), female = numeric(5))
for(i in 1:5){
  df_brier = mice::complete(imp_2107,i)
  bs = brier_score(df_brier, 1) #brier score
  bs.nhanes$male[i] = bs
  bs = brier_score(df_brier, 2) #brier score
  bs.nhanes$female[i] = bs
}

bs.fram = data.frame(male = numeric(5), female = numeric(5))
for(i in 1:5){
  df_brier = framingham_df
  bs = brier_score(df_brier, 1) #brier score
  bs.fram$male[i] = bs
  bs = brier_score(df_brier, 2) #brier score
  bs.fram$female[i] = bs
}
bs.fram = bs.fram[1,]
bs.nhanes = colMeans(bs.nhanes)
comparison_table <- data.frame(
  Population = c("NHANES", "Framingingham"),
  Male_Score = as.numeric(c(bs.nhanes[1], bs.fram[1])),
  Female_Score = as.numeric(c(bs.nhanes[2], bs.fram[2]))
)
comparison_table %>%
  kable("html") %>%
  kable_styling(full_width = FALSE)
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")

cov_mat = cov(framingham_df[,common_columns[c(3:5,9:10)]])

n_sim = 100
sd_sim = seq(0.1, 3, length.out = n_sim)
bs.df = data.frame(male = numeric(n_sim), female = numeric(n_sim))
error_vec = seq(0.1, 2, length.out = n_sim)
for(i in 1:n_sim){
    n_row = 1000
    error = error_vec[i]
    cov_mat = cov(framingham_df[,common_columns[c(3:5,9:10)]])*error
    mean_vec =  as.numeric(lapply(df_2017[,common_columns[c(3:5,9:10)]], mean, na.rm = TRUE))
    #conside associations with normal distribution
    sim_dat.cont = data.frame(mvrnorm(n = n_row, mu = mean_vec, Sigma = cov_mat))
    #binaryvar: CURSMOKE, BP_MED, SEX,DIABETES
    SEX = rbinom(n_row, size = 1, prob = 0.49)+1
    cor_mat = cor(framingham_df[,c("CURSMOKE", "BPMEDS","DIABETES")])
    cor_mat[lower.tri(cor_mat)] <-cor_mat[lower.tri(cor_mat)]*error
    cor_mat[upper.tri(cor_mat)] <- cor_mat[upper.tri(cor_mat)]*error
    BIN_VAR <- rmvbin(n = n_row, margprob = c(mean(df_2017$CURSMOKE, na.rm = TRUE), mean(df_2017$BPMEDS
    simulated_bin <- data.frame(CURSMOKE = BIN_VAR[, 1], BPMEDS = BIN_VAR[,2], DIABETES = BIN_VAR[,3], S
    sim_dat = cbind(sim_dat.cont, simulated_bin)


    bs = brier_score(sim_dat, 1) #brier score
    bs.df$male[i] = bs
    bs = brier_score(sim_dat, 2) #brier score
    bs.df$female[i] = bs

}
num_cuts = 7
calib_data.bs=data.frame(bs.df,
                         bin = cut(error_vec, breaks = num_cuts))
se <- calib_data.bs %>%
             group_by(bin) %>%
             dplyr::summarize(se.male = sqrt(var(male)/n()*(n()-1)),
                              se.female = sqrt(var(female)/n()*(n()-1)),
                              mean_male = median(male),
                              mean_female = median(female))
calib_data.bs <- left_join(calib_data.bs, se, by = "bin")

#male plot
ggplot(calib_data.bs, aes(x = bin)) +
  geom_point(aes(y = male), color = "blue", size = 1) +
  geom_errorbar(aes(ymin = mean_male - 1.96*se.male, ymax = mean_male + 1.96*se.male),width = 0.2, colo
  labs(title = "Male Brier Scores",
       x = "Association Scale",
       y = "Brier Score")+  theme_minimal()


#female sim plot
ggplot(calib_data.bs, aes(x = bin)) +
  geom_point(aes(y = female), color = "pink", size = 1) +
  geom_errorbar(aes(ymin = mean_female - 1.96*se.female, ymax = mean_female + 1.96*se.female),width = 0
  labs(title = "Female Brier Scores",
       x = "Association Scale",
       y = "Brier Score") +
  theme_minimal()

#Simulate over different number of rows
```

```r
summary_strat = df_2017[,-1] %>%
  tbl_summary(statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} / {N} ({p}%)"
    ))



sim_data.T.3 = sim_dat
sim_data.T.3$SEX = factor(sim_data.T.3$SEX, levels = c(1, 2), 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
#male plot
male_p = ggplot(calib_data.bs, aes(x = bin)) +
  geom_point(aes(y = male), color = "blue", size = 1) +
  geom_errorbar(aes(ymin = mean_male - 1.96*se.male, ymax = mean_male + 1.96*se.male),width = 0.2, colo
  labs(title = "Male Brier Scores",
       x = "Association Scale",
       y = "Brier Score")+  theme_minimal()


#female sim plot
female_p = ggplot(calib_data.bs, aes(x = bin)) +
  geom_point(aes(y = female), color = "pink", size = 1) +
  geom_errorbar(aes(ymin = mean_female - 1.96*se.female, ymax = mean_female + 1.96*se.female),width = 0
  labs(title = "Female Brier Scores",
       x = "Association Scale",
       y = "Brier Score") +
  theme_minimal()
male_p
female_p

library(DT)
se.mean = calib_data.bs %>% group_by(bin)%>%
  summarise(across(everything(), mean))
colnames(se.mean) = c("Association_interval","Male_Brier_Scores", "Female_Brier_Scores", "Male_SE", "Fem
kbl(se.mean[,1:5],booktabs = T)
```