Framingham Heart Study Notebook

The dataset consists of 11627 observations of 39 variables. We need to preprocess the data before building the model.

We start by identifying the subset of the dataframe such that each individual has only one observation corresponding to PERIOD = 1 and is free of CHD at the the time (PREVCHD = 0).

```
library(caTools)
framing <- read.csv("framingham.csv")
str(framing)
```

```
## 'data.frame':    11627 obs. of  39 variables:
##  $ RANDID  : int  2448 2448 6238 6238 6238 9428 9428 10552 10552 11252 ...
##  $ SEX     : int  1 1 2 2 2 1 1 2 2 2 ...
##  $ TOTCHOL : int  195 209 250 260 237 245 283 225 232 285 ...
##  $ AGE     : int  39 52 46 52 58 48 54 61 67 46 ...
##  $ SYSBP   : num  106 121 121 105 108 ...
##  $ DIABP   : num  70 66 81 69.5 66 80 89 95 109 84 ...
##  $ CURSMOKE: int  0 0 0 0 0 1 1 1 1 1 ...
##  $ CIGPDAY : int  0 0 0 0 0 20 30 30 20 23 ...
##  $ BMI     : num  27 NA 28.7 29.4 28.5 ...
##  $ DIABETES: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BPMEDS  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HEARTRTE: int  80 69 95 80 80 75 75 65 60 85 ...
##  $ GLUCOSE : int  77 92 76 86 71 70 87 103 89 85 ...
##  $ educ    : int  4 4 2 2 2 1 1 3 3 3 ...
##  $ PREVCHD : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVAP  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVMI  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVSTRK: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVHYP : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ TIME    : int  0 4628 0 2156 4344 0 2199 0 1977 0 ...
##  $ PERIOD  : int  1 3 1 2 3 1 2 1 2 1 ...
##  $ HDLC    : int  NA 31 NA NA 54 NA NA NA NA NA ...
##  $ LDLC    : int  NA 178 NA NA 141 NA NA NA NA NA ...
##  $ DEATH   : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ ANGINA  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HOSPMI  : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ MI_FCHD : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ ANYCHD  : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ STROKE  : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ CVD     : int  1 1 0 0 0 0 0 1 1 0 ...
##  $ HYPERTEN: int  0 0 0 0 0 0 0 1 1 1 ...
##  $ TIMEAP  : int  8766 8766 8766 8766 8766 8766 8766 2956 2956 8766 ...
##  $ TIMEMI  : int  6438 6438 8766 8766 8766 8766 8766 2956 2956 8766 ...
##  $ TIMEMIFC: int  6438 6438 8766 8766 8766 8766 8766 2956 2956 8766 ...
##  $ TIMECHD : int  6438 6438 8766 8766 8766 8766 8766 2956 2956 8766 ...
```

```
##  $ TIMESTRK: int  8766 8766 8766 8766 8766 8766 8766 2089 2089 8766 ...
##  $ TIMECVD : int  6438 6438 8766 8766 8766 8766 8766 2089 2089 8766 ...
##  $ TIMEDTH : int  8766 8766 8766 8766 8766 8766 8766 2956 2956 8766 ...
##  $ TIMEHYP : int  8766 8766 8766 8766 8766 8766 8766 0 0 4285 ...
```

```
head(framing)
```

```
##   RANDID SEX TOTCHOL AGE SYSBP DIABP CURSMOKE CIGPDAY   BMI DIABETES BPMEDS
## 1   2448   1     195  39 106.0  70.0        0       0 26.97        0      0
## 2   2448   1     209  52 121.0  66.0        0       0    NA        0      0
## 3   6238   2     250  46 121.0  81.0        0       0 28.73        0      0
## 4   6238   2     260  52 105.0  69.5        0       0 29.43        0      0
## 5   6238   2     237  58 108.0  66.0        0       0 28.50        0      0
## 6   9428   1     245  48 127.5  80.0        1      20 25.34        0      0
##   HEARTRTE GLUCOSE educ PREVCHD PREVAP PREVMI PREVSTRK PREVHYP TIME PERIOD HDLC
## 1       80      77    4       0      0      0        0       0    0      1   NA
## 2       69      92    4       0      0      0        0       0 4628      3   31
## 3       95      76    2       0      0      0        0       0    0      1   NA
## 4       80      86    2       0      0      0        0       0 2156      2   NA
## 5       80      71    2       0      0      0        0       0 4344      3   54
## 6       75      70    1       0      0      0        0       0    0      1   NA
##   LDLC DEATH ANGINA HOSPMI MI_FCHD ANYCHD STROKE CVD HYPERTEN TIMEAP TIMEMI
## 1   NA     0      0      1       1      1      0   1        0   8766   6438
## 2  178     0      0      1       1      1      0   1        0   8766   6438
## 3   NA     0      0      0       0      0      0   0        0   8766   8766
## 4   NA     0      0      0       0      0      0   0        0   8766   8766
## 5  141     0      0      0       0      0      0   0        0   8766   8766
## 6   NA     0      0      0       0      0      0   0        0   8766   8766
##   TIMEMIFC TIMECHD TIMESTRK TIMECVD TIMEDTH TIMEHYP
## 1     6438    6438     8766    6438    8766    8766
## 2     6438    6438     8766    6438    8766    8766
## 3     8766    8766     8766    8766    8766    8766
## 4     8766    8766     8766    8766    8766    8766
## 5     8766    8766     8766    8766    8766    8766
## 6     8766    8766     8766    8766    8766    8766
```

We start by identifying the subset of the dataframe such that each individual has only one observation corresponding to PERIOD = 1 and is free of CHD at the the time (PREVCHD = 0).

```
framing1 <- subset(x = framing, PERIOD == 1 & PREVCHD == 0)
str(framing1)
```

```
## 'data.frame':    4240 obs. of  39 variables:
##  $ RANDID  : int  2448 6238 9428 10552 11252 11263 12629 12806 14367 16365 ...
##  $ SEX     : int  1 2 1 2 2 2 2 2 1 1 ...
##  $ TOTCHOL : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ AGE     : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ SYSBP   : num  106 121 128 150 130 ...
##  $ DIABP   : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ CURSMOKE: int  0 0 1 1 1 0 0 1 0 1 ...
##  $ CIGPDAY : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BMI     : num  27 28.7 25.3 28.6 23.1 ...
```

```
##  $ DIABETES: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BPMEDS  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HEARTRTE: int  80 95 75 65 85 77 60 79 76 93 ...
##  $ GLUCOSE : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ educ    : int  4 2 1 3 3 2 1 2 1 1 ...
##  $ PREVCHD : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVAP  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVMI  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVSTRK: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVHYP : int  0 0 0 1 0 1 0 0 1 1 ...
##  $ TIME    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PERIOD  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ HDLC    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ LDLC    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ DEATH   : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ ANGINA  : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ HOSPMI  : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ MI_FCHD : int  1 0 0 0 0 1 0 0 0 0 ...
##  $ ANYCHD  : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ STROKE  : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ CVD     : int  1 0 0 1 0 1 0 0 0 0 ...
##  $ HYPERTEN: int  0 0 0 1 1 1 1 1 1 1 ...
##  $ TIMEAP  : int  8766 8766 8766 2956 8766 8766 373 8766 8766 8766 ...
##  $ TIMEMI  : int  6438 8766 8766 2956 8766 8766 8766 8766 8766 8766 ...
##  $ TIMEMIFC: int  6438 8766 8766 2956 8766 5719 8766 8766 8766 8766 ...
##  $ TIMECHD : int  6438 8766 8766 2956 8766 5719 373 8766 8766 8766 ...
##  $ TIMESTRK: int  8766 8766 8766 2089 8766 8766 8766 8766 8766 8766 ...
##  $ TIMECVD : int  6438 8766 8766 2089 8766 5719 8766 8766 8766 8766 ...
##  $ TIMEDTH : int  8766 8766 8766 2956 8766 8766 8766 8766 8766 8766 ...
##  $ TIMEHYP : int  8766 8766 8766 0 4285 0 2212 8679 0 0 ...
```

The new dataframe consists of 4240 observations of 39 variables.

To model the event data, we need to identify patients if they had CHD in 10 years from their first visit. We do so by converting time for CHD to years (roughly) and check if CHD occurs in 10 years. The maximum range of the data in years is 24 in this dataset.

```
# currently the unit is in days
framing1$TENCHD <- as.integer((framing1$TIMECHD / 365) <= 10)
table(framing1$TENCHD)
```

```
##
##    0    1
## 3596  644
```

```
# 144 people have CHD in the next ten years
```

We work with some of the more relevant variables in this dataset for our purposes of prediction.

```
colnames(framing1)
```

```
##  [1] "RANDID"   "SEX"      "TOTCHOL"  "AGE"      "SYSBP"    "DIABP"
##  [7] "CURSMOKE" "CIGPDAY"  "BMI"      "DIABETES" "BPMEDS"   "HEARTRTE"
```

```
## [13] "GLUCOSE"  "educ"     "PREVCHD"  "PREVAP"   "PREVMI"   "PREVSTRK"
## [19] "PREVHYP"  "TIME"     "PERIOD"   "HDLC"     "LDLC"     "DEATH"
## [25] "ANGINA"   "HOSPMI"   "MI_FCHD"  "ANYCHD"   "STROKE"   "CVD"
## [31] "HYPERTEN" "TIMEAP"   "TIMEMI"   "TIMEMIFC" "TIMECHD"  "TIMESTRK"
## [37] "TIMECVD"  "TIMEDTH"  "TIMEHYP"  "TENCHD"
```

```
which(colnames(framing1) == "PERIOD")
```

```
## [1] 21
```

```
framing1 <- framing1[, c(1:21, 40)]
str(framing1)
```

```
## 'data.frame':    4240 obs. of  22 variables:
##  $ RANDID  : int  2448 6238 9428 10552 11252 11263 12629 12806 14367 16365 ...
##  $ SEX     : int  1 2 1 2 2 2 2 2 1 1 ...
##  $ TOTCHOL : int  195 250 245 225 285 228 205 313 260 225 ...
##  $ AGE     : int  39 46 48 61 46 43 63 45 52 43 ...
##  $ SYSBP   : num  106 121 128 150 130 ...
##  $ DIABP   : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ CURSMOKE: int  0 0 1 1 1 0 0 1 0 1 ...
##  $ CIGPDAY : int  0 0 20 30 23 0 0 20 0 30 ...
##  $ BMI     : num  27 28.7 25.3 28.6 23.1 ...
##  $ DIABETES: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ BPMEDS  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HEARTRTE: int  80 95 75 65 85 77 60 79 76 93 ...
##  $ GLUCOSE : int  77 76 70 103 85 99 85 78 79 88 ...
##  $ educ    : int  4 2 1 3 3 2 1 2 1 1 ...
##  $ PREVCHD : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVAP  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVMI  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVSTRK: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PREVHYP : int  0 0 0 1 0 1 0 0 1 1 ...
##  $ TIME    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PERIOD  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TENCHD  : int  0 0 0 1 0 0 1 0 0 0 ...
```

The variables in the new dataframe include RANDID (identification number), SEX (1=M,2=F), TOTCHOL (total cholesterol in mg/dL), AGE (age at the exam in years), SYSBP (systolic blood pressure), DIABP (diastolic blood pressure), CURSMOKE (current smoking 0=N,1=Y), CIGPDAY (cigarettes per day), BMI (body mass index), DIABETES (0=Not diabetic,1=Diabetic), BPMEDS (use of antihypertensive medication 1=YES,0=NO), HEARTRTE (heart rate in beats/min), GLUCOSE (glucose in mg/dL), educ (1=0-11 years,2=High school,3=Some college or vocational,4=College or more), PREVCHD(0=free of disease,1=prevalent disease), PREVAP (prevalent agina pectoris 1=YES,0=NO), PREVMI (prevalent myocardial infection 1=YES,0=NO), PREVSTRK (prevalent stroke 1=YES,0=NO), TIME (number of days since examination - all 0 here as it is the first exam date), PERIOD (Time period). We will use these variables to predict TENCHD. Note that when we use data from secondary sources, we often need to do some preprocessing before we can apply quantitative models to it.

Split the dataset into a training and test set. We will use the training set to build the model and the test set to simply check how the model does. We do so by preserving the ratios of the outcome variable in the two sets which can be done here. The caTools package can help to do so with the sample.split function. We set a seed to make the results replicable across different computers. Note that the split maintains the balance of people with CHD and without CHD in both the training and test sets.

```r
library(caTools)
?sample.split
set.seed(1)
# split data in the manner that train and test sets have similar proportion of 1 and 0 true labels
# returns values of TRUE and FALSE for each observation, to separate the dataset into TRUE and FALSE gr
split <- sample.split(Y = framing1$TENCHD, SplitRatio = 0.65)  # 65% in train set, remaining in test se
# split

# all values split with TRUE value goes to the train set
training <- subset(x = framing1, split == TRUE)
test <- subset(x = framing1, split == FALSE)

mean(training$TENCHD)
```

```
## [1] 0.1520319
```

```r
mean(test$TENCHD)
```

```
## [1] 0.1516173
```

Develop a logistic regression model. The use of ~. lets us use all the variables (other than TENCHD) in making a prediction. The results indicate that variables such as RANDID do not play a role. This is to be expected. EDUC is also not significant which seems reasonable though there might be a counter argument that if a person is more education, they give greater importance to health. We will leave the variable in and rebuild the model.

```r
model1 <- glm(formula = TENCHD ~ ., data = training, family = "binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = TENCHD ~ ., family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2162  -0.6057  -0.4474  -0.3029   2.7904
##
## Coefficients: (5 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.874e+00  8.700e-01  -7.902 2.75e-15 ***
## RANDID      -3.594e-09  2.061e-08  -0.174 0.861547
## SEX         -5.471e-01  1.331e-01  -4.111 3.94e-05 ***
## TOTCHOL      1.887e-03  1.376e-03   1.371 0.170246
## AGE          5.475e-02  8.083e-03   6.773 1.26e-11 ***
## SYSBP        1.231e-02  4.684e-03   2.628 0.008601 **
## DIABP       -2.138e-03  7.979e-03  -0.268 0.788704
## CURSMOKE    -1.219e-01  1.943e-01  -0.628 0.530190
## CIGPDAY      2.168e-02  7.823e-03   2.771 0.005584 **
## BMI          1.234e-02  1.526e-02   0.809 0.418782
## DIABETES    -1.934e-01  3.897e-01  -0.496 0.619678
## BPMEDS       7.219e-02  3.027e-01   0.238 0.811495
## HEARTRTE    -1.251e-03  5.104e-03  -0.245 0.806357
```

```
## GLUCOSE      1.034e-02  3.056e-03   3.384 0.000715 ***
## educ        -4.412e-02  6.049e-02  -0.729 0.465841
## PREVCHD             NA         NA      NA       NA
## PREVAP              NA         NA      NA       NA
## PREVMI              NA         NA      NA       NA
## PREVSTRK    -2.027e-01  6.845e-01  -0.296 0.767168
## PREVHYP      2.609e-01  1.669e-01   1.563 0.117958
## TIME                NA         NA      NA       NA
## PERIOD              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2037.5  on 2383  degrees of freedom
## Residual deviance: 1834.1  on 2367  degrees of freedom
##   (372 observations deleted due to missingness)
## AIC: 1868.1
##
## Number of Fisher Scoring iterations: 5
```

```r
# remove the variables with NA componenets based on Model 1 (ie. cleaning the model)
model2 <- glm(formula = TENCHD ~ SEX + TOTCHOL + AGE + SYSBP + DIABP + CIGPDAY + CURSMOKE + BMI + DIABE
summary(model2)
```

```
##
## Call:
## glm(formula = TENCHD ~ SEX + TOTCHOL + AGE + SYSBP + DIABP +
##     CIGPDAY + CURSMOKE + BMI + DIABETES + BPMEDS + HEARTRTE +
##     GLUCOSE + educ + PREVSTRK + PREVHYP, family = "binomial",
##     data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2221  -0.6061  -0.4469  -0.3030   2.7865
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.893440   0.863061  -7.987 1.38e-15 ***
## SEX         -0.546744   0.133067  -4.109 3.98e-05 ***
## TOTCHOL      0.001892   0.001376   1.375 0.168999
## AGE          0.054723   0.008082   6.771 1.28e-11 ***
## SYSBP        0.012335   0.004682   2.635 0.008415 **
## DIABP       -0.002156   0.007979  -0.270 0.787031
## CIGPDAY      0.021662   0.007823   2.769 0.005625 **
## CURSMOKE    -0.121615   0.194276  -0.626 0.531322
## BMI          0.012397   0.015259   0.812 0.416545
## DIABETES    -0.195542   0.389503  -0.502 0.615646
## BPMEDS       0.070097   0.302526   0.232 0.816767
## HEARTRTE    -0.001282   0.005101  -0.251 0.801555
## GLUCOSE      0.010346   0.003058   3.383 0.000716 ***
## educ        -0.044207   0.060493  -0.731 0.464913
## PREVSTRK    -0.204484   0.684401  -0.299 0.765109
## PREVHYP      0.259538   0.166680   1.557 0.119446
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2037.5  on 2383  degrees of freedom
## Residual deviance: 1834.1  on 2368  degrees of freedom
##   (372 observations deleted due to missingness)
## AIC: 1866.1
##
## Number of Fisher Scoring iterations: 5
```

The result indicate that the variables signficant at the 0.001 level are the SEX, AGE, SYSBP, CIGPDAY, GLUCOSE (and the intercept). Suppose we use only these variables and refit the model.

```
model3 <- glm(formula = TENCHD ~ SEX + AGE + SYSBP + CIGPDAY + GLUCOSE, data = training, family = "bino
summary(model3)
```

```
##
## Call:
## glm(formula = TENCHD ~ SEX + AGE + SYSBP + CIGPDAY + GLUCOSE,
##     family = "binomial", data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3252  -0.5973  -0.4480  -0.3089   2.7699
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.160110   0.524258 -13.658  < 2e-16 ***
## SEX         -0.540547   0.126240  -4.282 1.85e-05 ***
## AGE          0.059845   0.007537   7.940 2.02e-15 ***
## SYSBP        0.016072   0.002573   6.247 4.18e-10 ***
## CIGPDAY      0.018223   0.005058   3.602 0.000315 ***
## GLUCOSE      0.009935   0.002303   4.314 1.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2143.5  on 2496  degrees of freedom
## Residual deviance: 1929.3  on 2491  degrees of freedom
##   (259 observations deleted due to missingness)
## AIC: 1941.3
##
## Number of Fisher Scoring iterations: 5
```

All the variables are significant in this new model. While the AIC is more as compared to model 2, in this example let us stick with model 3 for interpretability reasons as it has fewer variables.

Suppose we have a male who is 60 years old with a systolic blood pressure of 145, smokes two cigarettes per day with a glucose level of 80. For this patient, we can predict the probability of getting a 10 year CHD as roughly 0.279.

```r
model3$coefficients
```

```
##  (Intercept)          SEX          AGE        SYSBP      CIGPDAY      GLUCOSE
## -7.160109865 -0.540546500  0.059844683  0.016071917  0.018222649  0.009934894
```

```r
log_odds <- t(model3$coefficients) %*% c(1, 1, 60, 145, 2, 80)
log_odds
```

```
##            [,1]
## [1,] -0.9483106
```

```r
probs <- exp(log_odds) / (1 + exp(log_odds))
probs
```

```
##           [,1]
## [1,] 0.2792247
```

Prediction

```r
predict_test <- predict(model3, newdata = test, type = "response")
# predict_test

# create confusion matrix
# threshold = 0.5
CM <- table(predict_test > 0.5, test$TENCHD)  # check with lower values

CM
```

```
##
##             0    1
##   FALSE 1113  187
##   TRUE     9   21
```

```r
Accuracy = (CM[1,1] + CM[2,2]) / sum(CM)
Accuracy
```

```
## [1] 0.8526316
```

```r
BaseAccuracy =  (sum(CM[1:2,1])) / sum(CM)
BaseAccuracy
```

```
## [1] 0.843609
```

```r
Specificity = (CM[1,1]) / sum(CM[1:2,1])
Sensitivity = (CM[2,2]) / sum(CM[1:2,2])

Specificity
```

```
## [1] 0.9919786
```

```
Sensitivity
```

```
## [1] 0.1009615
```

The accuracy in the test set is 0.8526. We can compare this to a baseline model which predicts no one has CHD. The accuracy of the baseline model is 0.8436. Note that we have fewer observations in the test set than the full set on which we make predictions as there are some missing entries in the dataframe.

```
table(training$TENCHD)
```

```
##
##    0    1
## 2337  419
```

```
# check with lower values 0.25
CM2 <- table(predict_test > 0.25, test$TENCHD)
CM2
```

```
##
##           0   1
##   FALSE 987 121
##   TRUE  135  87
```

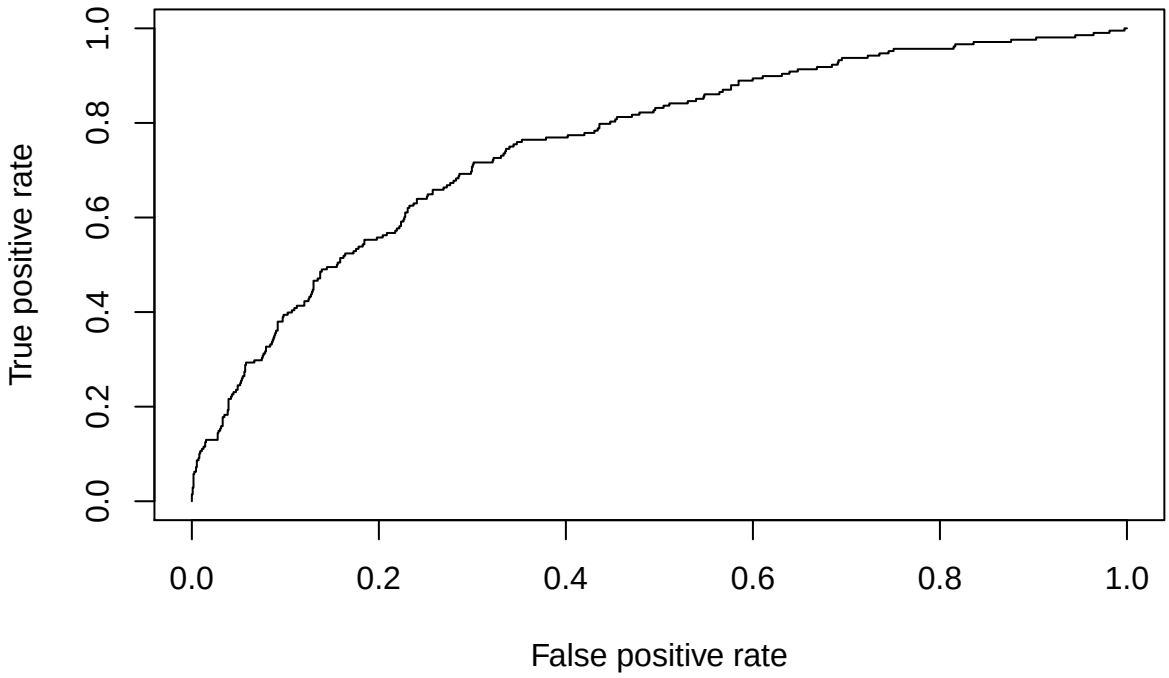
Using a threshold of 0.5, the model beats the baseline model by a small amount. Suppose we use a lower threshold. We are prone to more false positives as we will be predicting more people to have CHD but this might be reasonable for this application as compared to obtaining more false negatives. While more resources might be spent on unnecessary patients, the cost of death versus preventive decisions is a trade-off to be made here. The effects of false negatives are much more than false positive here. If clinicians used this model, then 222 (135+87) patients would be suggested some treatment, of which 135 would be unnecessary. Accuracy might not always be the most important measure in certain applications as this example illustrates.

More details: Plotting and ROCR

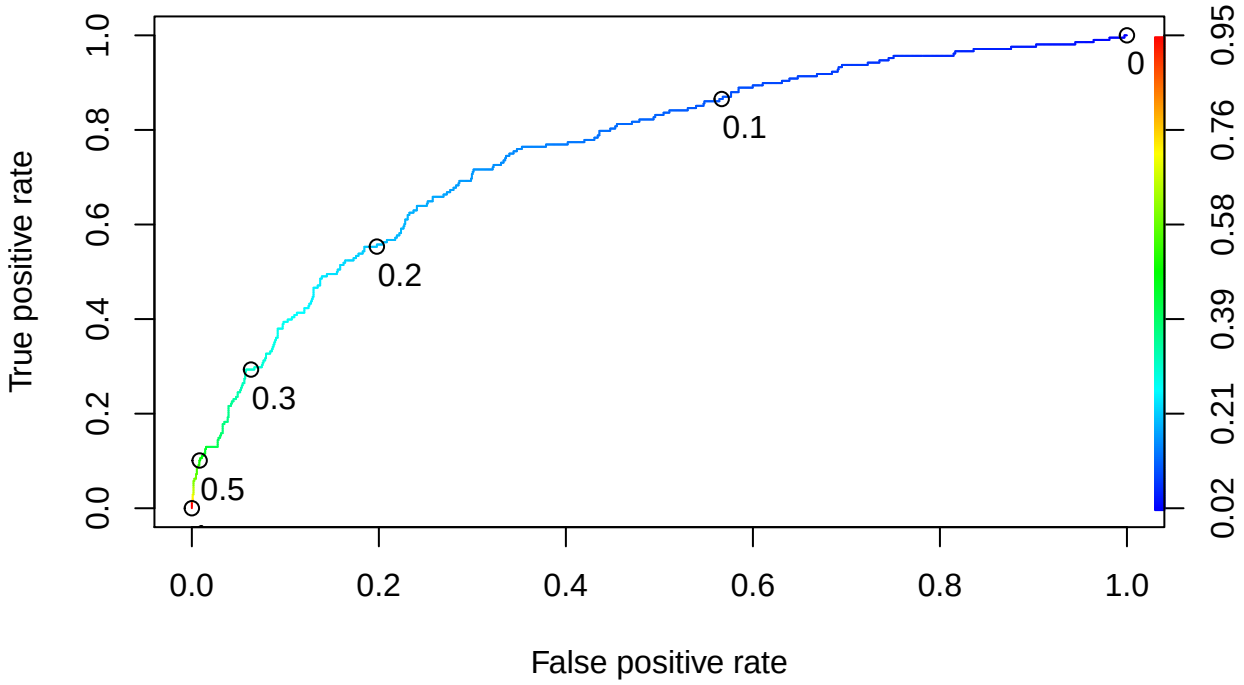
```
library(ROCR)
####  predict3 <- prediction(predict_test,test$TENCHD) ... should work but due to NA..

# predict only on values that have no NA values across the input data
pr3 <- complete.cases(cbind(predict_test, test$TENCHD))
predict3 <- prediction(predict_test[pr3], test$TENCHD[pr3])
perf3 <- performance(predict3, x.measure = "fpr", measure = "tpr")

plot(perf3)
```




```
plot(perf3, colorize = T, print.cutoffs.at = c(0, 0.1, 0.2, 0.3, 0.5, 1), text.adj = c(-.02, 1.7))
```




```
as.numeric(performance(predict3, measure = "auc")@y.values)
```

```
## [1] 0.7574258
```

```
#summary(model3)
```

The result indicates that the model can distinguish between low and high risk patients better than random guessing.