432 Class 10

https://thomaselove.github.io/432-2024

2024-02-15

## Today’s Agenda

Fitting and evaluating logistic regression models with lrm

* The Framingham example
  + Outcome: chd10 = Developed coronary heart disease in next 10 years?
* Use lrm to model chd10 using four predictors
  + on the complete cases (fram\_cc)
  + accounting for missingness via single imputation
  + accounting for missingness via multiple imputation
* Consider adding non-linear terms, refit and re-evaluate

## Today’s R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(cutpointr) ## NEW: specifies "optimal" cutpoints  
library(caret) ## for creating a confusion matrix  
library(pROC) ## should come after cutpointr  
library(ROCR) ## NEW: alternative to pROC for plotting ROC curves  
library(janitor)  
library(broom)  
library(naniar)  
library(mice) ## we'll use for single imputation today  
library(rms) ## also (automatically) loads Hmisc  
library(tidyverse)  
  
theme\_set(theme\_bw())

# The “Framingham” Data

## The Data

fram\_raw <- read\_csv("c10/data/framingham.csv",   
 show\_col\_types = FALSE) |>  
 clean\_names()

See <https://www.framinghamheartstudy.org/> for more details.

* The variables describe n = 4238 adults examined at baseline, then followed for 10 years to see if they developed incident coronary heart disease.
* This particular data set is purportedly from the Framingham study.

## Today’s Six Variables

Data management for these variables shown in next slide.

| Variable | Description |
| --- | --- |
| subj\_id | identifying code added by Dr. Love |
| chd10 | 1 = coronary heart disease in next 10 years, else 0 |
| educ | four-level factor: educational attainment |
| glucose | blood glucose level in mg/dl |
| sbp | systolic blood pressure (mm Hg) |
| smoker | 1 = current smoker at time of examination, else 0 |

## Data Cleanup

fram\_orig <- fram\_raw |>  
 mutate(educ =   
 fct\_recode(factor(education),   
 "Some HS" = "1",  
 "HS grad" = "2",  
 "Some Coll" = "3",  
 "Coll grad" = "4")) |>  
 rename(smoker = "current\_smoker",  
 cigs = "cigs\_per\_day",  
 stroke = "prevalent\_stroke",  
 highbp = "prevalent\_hyp",  
 chol = "tot\_chol",  
 sbp = "sys\_bp", dbp = "dia\_bp",  
 hrate = "heart\_rate",  
 chd10 = "ten\_year\_chd") |>  
 select(subj\_id, chd10, educ, glucose, sbp, smoker,  
 everything()) |> select(-education)

## Other 11 variables in fram\_orig

| Variable | Description |
| --- | --- |
| male | 1 = subject is male, else 0 |
| age | in years (range is 32 to 70) |
| cigs | number of cigarettes smoked per day |
| bp\_meds | 1 = using anti-hypertensive medication |
| stroke | 1 = history of stroke, else 0 |
| highbp | 1 = under treatment for hypertension, else 0 |
| diabetes | 1 = history of diabetes, else 0 |
| chol | total cholesterol (mg/dl) |
| dbp | diastolic blood pressure (mm Hg) |
| bmi | body mass index in |
| hrate | heart rate in beats per minute |

## Missing Data?

Our outcome chd10 has no missing values.

fram\_orig |> tabyl(chd10) |> adorn\_pct\_formatting(digits = 1)

chd10 n percent  
 0 3594 84.8%  
 1 644 15.2%

* 3656 (86.3%) of the 4238 subjects in fram\_orig are complete.
* The remaining 582 observations have something missing.

n\_case\_complete(fram\_orig); pct\_complete\_case(fram\_orig)

[1] 3656

[1] 86.26711

## Counts of Missing Data, by Variable

miss\_var\_summary(fram\_orig) |>   
 filter(n\_miss > 0)

# A tibble: 7 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <dbl>  
1 glucose 388 9.16   
2 educ 105 2.48   
3 bp\_meds 53 1.25   
4 chol 50 1.18   
5 cigs 29 0.684   
6 bmi 19 0.448   
7 hrate 1 0.0236

While the only four predictors we’ll use today for chd10 are educ, glucose, sbp and smoke, we’ll impute all of the missing values, using the complete set of 17 variables.

## Imputation via mice

We need to impute:

* 5 quantities (glucose, bmi, cigs, chol and hrate)
* 1 binary variable (bp\_meds), and
* 1 multi-categorical variable (educ)

We have missing data in 13.7% of our observations, so we’ll use mice to create 15 imputed data sets, and then save one of them as our “single imputation” tibble.

set.seed(432432)  
fram\_mice15 <- mice(fram\_orig, m = 15, printFlag = FALSE)

## Store 12th imputation as fram\_si

fram\_si <- complete(fram\_mice15, 12) |> tibble()  
  
n\_miss(fram\_si)

[1] 0

fram\_si

# A tibble: 4,238 × 17  
 subj\_id chd10 educ glucose sbp smoker male age cigs bp\_meds stroke  
 <dbl> <dbl> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 1 0 Coll grad 77 106 0 1 39 0 0 0  
 2 2 0 HS grad 76 121 0 0 46 0 0 0  
 3 3 0 Some HS 70 128. 1 1 48 20 0 0  
 4 4 1 Some Coll 103 150 1 0 61 30 0 0  
 5 5 0 Some Coll 85 130 1 0 46 23 0 0  
 6 6 0 HS grad 99 180 0 0 43 0 0 0  
 7 7 1 Some HS 85 138 0 0 63 0 0 0  
 8 8 0 HS grad 78 100 1 0 45 20 0 0  
 9 9 0 Some HS 79 142. 0 1 52 0 0 0  
10 10 0 Some HS 88 162 1 1 43 30 0 0  
# ℹ 4,228 more rows  
# ℹ 6 more variables: highbp <dbl>, diabetes <dbl>, chol <dbl>, dbp <dbl>,  
# bmi <dbl>, hrate <dbl>

## Check multi-categorical imputation?

fram\_orig |> tabyl(educ) |> adorn\_pct\_formatting()

educ n percent valid\_percent  
 Some HS 1720 40.6% 41.6%  
 HS grad 1253 29.6% 30.3%  
 Some Coll 687 16.2% 16.6%  
 Coll grad 473 11.2% 11.4%  
 <NA> 105 2.5% -

fram\_si |> tabyl(educ) |> adorn\_pct\_formatting()

educ n percent  
 Some HS 1772 41.8%  
 HS grad 1284 30.3%  
 Some Coll 702 16.6%  
 Coll grad 480 11.3%

Do the imputed values seem like reasonable choices?

## Data Sets for today’s analyses

fram\_start <- fram\_orig |>   
 select(subj\_id, chd10, glucose, smoker, sbp, educ)  
  
fram\_cc <- fram\_start |>  
 drop\_na()  
  
fram\_si <- fram\_si |>   
 select(subj\_id, chd10, glucose, smoker, sbp, educ)

* fram\_start contains 4238 rows and the 6 columns we’ll use, with 388 rows missing glucose and 105 missing educ.
* fram\_cc: (complete cases) includes only the 3753 complete rows for our 6 columns.
* fram\_si: singly imputed to yield 4238 rows on our 6 columns with complete data.

## Modeling Plan

Use lrm to fit a four-predictor logistic regression model to predict chd10 using glucose, smoker, sbp and educ

1. Using the complete cases (fram\_cc)
2. Accounting for missingness via single imputation (fram\_si)
3. Accounting for missingness via multiple imputation, via aregImpute()

Then, we’ll consider adding several non-linear terms to the “four-predictor” models, and refit.

# Fitting a Four-Predictor Model using Complete Cases

## A “Four Predictor” model

First, we’ll use the fram\_cc data to perform a complete-case analysis and fix ideas.

d <- datadist(fram\_cc)  
options(datadist = "d")  
  
mod\_cc <- lrm(chd10 ~ glucose + smoker + sbp + educ,  
 data = fram\_cc, x = TRUE, y = TRUE)

This works very nicely when chd10 = 1 (for Yes) or 0 (for No), as it does here. What if your outcome was actually a factor with values Yes and No? Use the following…

mod\_cc <- lrm((outcome == "Yes") ~   
 glucose + smoker + sbp + educ,  
 data = fram\_cc, x = TRUE, y = TRUE)

## Main Output for mod\_cc

mod\_cc

Logistic Regression Model  
  
lrm(formula = chd10 ~ glucose + smoker + sbp + educ, data = fram\_cc,   
 x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 3753 LR chi2 223.29 R2 0.100 C 0.682   
 0 3174 d.f. 6 R2(6,3753)0.056 Dxy 0.363   
 1 579 Pr(> chi2) <0.0001 R2(6,1469)0.137 gamma 0.364   
max |deriv| 2e-11 Brier 0.122 tau-a 0.095   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -5.5622 0.3217 -17.29 <0.0001   
glucose 0.0081 0.0016 4.93 <0.0001   
smoker 0.3126 0.0955 3.27 0.0011   
sbp 0.0237 0.0020 12.05 <0.0001   
educ=HS grad -0.4674 0.1157 -4.04 <0.0001   
educ=Some Coll -0.3924 0.1423 -2.76 0.0058   
educ=Coll grad -0.1356 0.1549 -0.88 0.3815

## Deconstructing mod\_cc summaries, 1

Logistic Regression Model  
lrm(formula = chd10 ~ glucose + smoker + sbp + educ, data = fram\_cc,   
 x = TRUE, y = TRUE)  
  
Obs = 3753 0 = 3174 1 = 579 max |deriv| 2e-11

* Obs = Observations used to fit model, with 0 = the # of zeros and 1 = the # of ones in our outcome, chd10.
* max |deriv| is the maximum absolute value of the derivative at the point where the maximum likelihood function was estimated.
  + All we care about is whether the iterative function-fitting process converged, and R will warn you if it doesn’t.

## Deconstructing mod\_cc summaries, 2

Model Likelihood Ratio Test: LR chi2 = 223.29, d.f. = 6 Pr(> chi2) <0.0001

* This is a global likelihood ratio test (drop in deviance test.)
* Likelihood Ratio = null deviance - residual deviance
  + d.f. = null d.f. - residual d.f., so mod\_cc uses 6 df.
* Pr(> chi2) is a *p* value obtained from comparison to a distribution with appropriate d.f.
  + The null hypothesis (that the model has no predictive value at all) is rarely of practical interest.

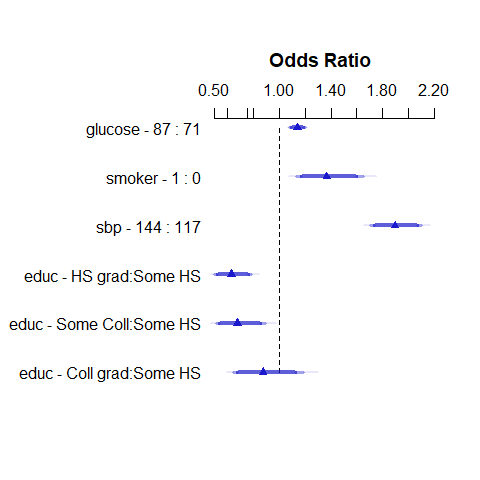
## Deconstructing mod\_cc summaries, 3

Coef S.E. Wald Z Pr(>|Z|)  
Intercept -5.5622 0.3217 -17.29 <0.0001   
glucose 0.0081 0.0016 4.93 <0.0001   
smoker 0.3126 0.0955 3.27 0.0011   
sbp 0.0237 0.0020 12.05 <0.0001   
educ=HS grad -0.4674 0.1157 -4.04 <0.0001   
educ=Some Coll -0.3924 0.1423 -2.76 0.0058   
educ=Coll grad -0.1356 0.1549 -0.88 0.3815

* How does each predictor appear to relate to 10-year risk?
  + Which is the baseline educ category?
  + Remember that these estimates are on the logit scale.

## Plot of Effects using mod\_cc

plot(summary(mod\_cc))



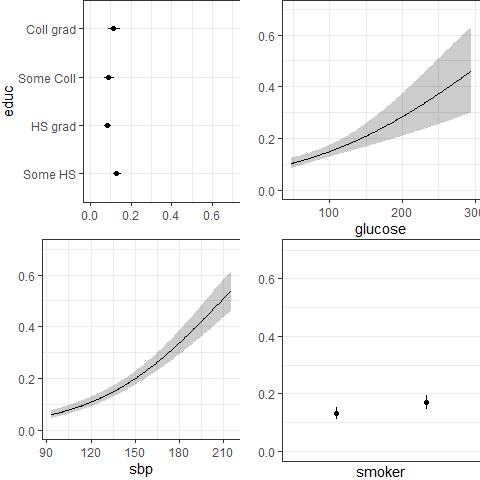
## Effect Size Summary for mod\_cc

summary(mod\_cc)

Effects Response : chd10   
  
 Factor Low High Diff. Effect S.E. Lower 0.95  
 glucose 71 87 16 0.12912 0.026171 0.077828   
 Odds Ratio 71 87 16 1.13780 NA 1.080900   
 smoker 0 1 1 0.31259 0.095453 0.125510   
 Odds Ratio 0 1 1 1.36700 NA 1.133700   
 sbp 117 144 27 0.63907 0.053053 0.535080   
 Odds Ratio 117 144 27 1.89470 NA 1.707600   
 educ - HS grad:Some HS 1 2 NA -0.46740 0.115720 -0.694220   
 Odds Ratio 1 2 NA 0.62663 NA 0.499470   
 educ - Some Coll:Some HS 1 3 NA -0.39238 0.142310 -0.671310   
 Odds Ratio 1 3 NA 0.67544 NA 0.511040   
 educ - Coll grad:Some HS 1 4 NA -0.13556 0.154910 -0.439180   
 Odds Ratio 1 4 NA 0.87323 NA 0.644570   
 Upper 0.95  
 0.18041   
 1.19770   
 0.49968   
 1.64820   
 0.74305   
 2.10230   
 -0.24059   
 0.78616   
 -0.11346   
 0.89274   
 0.16806   
 1.18300

## Predict results for mod\_cc

ggplot(Predict(mod\_cc, fun = plogis))



## Deconstructing mod\_cc summaries, 4

Discrimination Indexes Rank Discrimination Indexes   
R2 0.100 C 0.682   
R2(6,3753) 0.056 Dxy 0.363   
R2(6,1469) 0.137 gamma 0.364   
Brier 0.122 tau-a 0.095

The main things we’ll care about are:

* Nagelkerke , symbolized R2 here.
* The Brier score, symbolized Brier.
* The area under the ROC curve, or C statistic, shown as C.
* Somers’ d statistic, symbolized Dxy here.

Let’s walk through each of those, in turn.

## Key Indexes (Nagelkerke )

* The Nagelkerke reaches 1 if the fitted model shows as much improvement as possible over the null model (which just predicts the mean response on the 0-1 scale for all subjects).
* Nagelkerke is 0 for the null model, and is larger (closer to 1) as the fitted model improves, although it’s criticized for being misleadingly high,
* A Nagelkerke value of 0.100 doesn’t mean 10% of anything.

Here, Nagelkerke = 0.100 indicates fairly low quality of fit.

## An Alternative: McFadden’s

McFadden R-square = 1 minus the ratio of (the model deviance over the deviance for the null model.)

* To obtain this for our mod\_cc run with lrm, use:

1 - (mod\_cc$deviance[2] / mod\_cc$deviance[1])

[1] 0.069174

* This McFadden corresponds well to the proportionate reduction in error interpretation of an , if that’s all you need.

## Key Indexes (Brier Score = 0.122)

* The lower the Brier score, the better the predictions are calibrated.
* The maximum (worst) score is 1, the best is 0.

From Wikipedia: Suppose you forecast the probability P that it will rain tomorrow.

* If the forecast is P = 1 (100%) and it rains, the Brier Score is 0.
* If the forecast is P = 1 (100%) and it doesn’t rain, the Brier Score is 1.
* If the forecast is P = 0.7 and it rains, Brier = .
* If the forecast is P = 0.3 and it rains, Brier = .
* If the forecast is P = 0.5, the Brier score is regardless of whether it rains.

## Is collinearity a problem?

rms::vif(mod\_cc)

glucose smoker sbp educ=HS grad educ=Some Coll   
 1.011147 1.036391 1.047904 1.134669 1.106597   
educ=Coll grad   
 1.100638

## Receiver Operating Characteristic Curve Analysis

One way to assess the predictive accuracy within the model development sample in a logistic regression is to consider analyses based on the receiver operating characteristic (ROC) curve. ROC curves are commonly used in assessing diagnoses in medical settings, and in signal detection applications.

The accuracy of a test can be evaluated by considering two types of errors: false positives and false negatives.

## C = 0.682 (area under ROC curve)

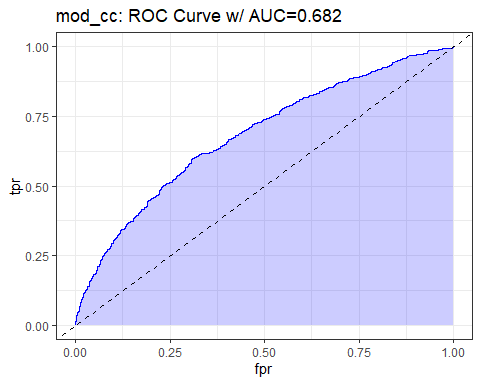
The C statistic and Somers’ d (Dxy) are connected:

The C statistic ranges from 0 to 1.

* C = 0.5 describes a prediction that is exactly as good as random guessing
* C = 1 indicates a perfect prediction model, one that guesses “yes” for all patients with chd10 = 1 and which guesses “no” for all patients with chd10 = 0.
* Most of the time, the closer to 1, the happier we are:
  + usually indicates a moderately strong model (good discrimination)
  + indicates a very strong model (excellent discrimination)

So 0.682 isn’t good.

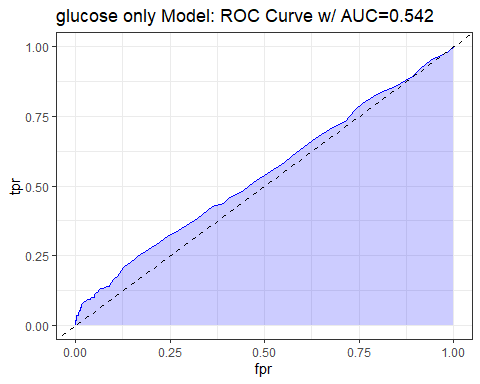
## ROC Curve for our mod\_cc



## Code for Previous Slide

## requires ROCR package  
prob <- predict(mod\_cc, type="fitted")  
pred <- prediction(prob, fram\_cc$chd10)  
perf <- performance(pred, measure = "tpr", x.measure = "fpr")  
auc <- performance(pred, measure="auc")  
  
auc <- round(auc@y.values[[1]],3)  
roc.data <- data.frame(fpr=unlist(perf@x.values),  
 tpr=unlist(perf@y.values),  
 model="GLM")  
  
ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +  
 geom\_ribbon(alpha=0.2, fill = "blue") +  
 geom\_line(aes(y=tpr), col = "blue") +  
 geom\_abline(intercept = 0, slope = 1, lty = "dashed") +  
 labs(title = paste0("Model A: ROC Curve w/ AUC=", auc))

## ROC Curve for glucose only model



## Validated Summaries for mod\_cc

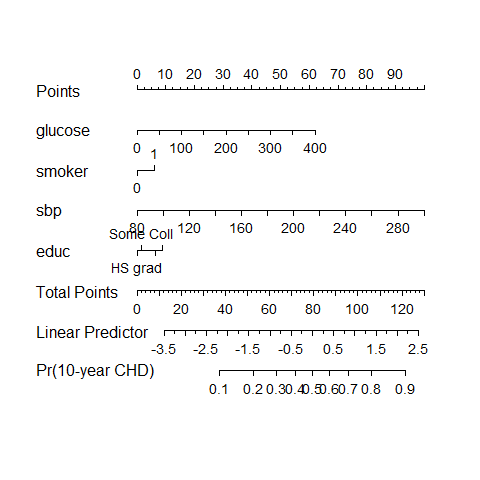
* Correcting for over-optimism through bootstrap validation, with 50 resamples.
* We’ll focus on C (recall C = 0.5 + Dxy/2), R2 and B (Brier).

set.seed(4321); validate(mod\_cc, B = 50)

index.orig training test optimism index.corrected n  
Dxy 0.3634 0.3659 0.3585 0.0073 0.3561 50  
R2 0.1001 0.1021 0.0978 0.0044 0.0958 50  
Intercept 0.0000 0.0000 -0.0274 0.0274 -0.0274 50  
Slope 1.0000 1.0000 0.9802 0.0198 0.9802 50  
Emax 0.0000 0.0000 0.0094 0.0094 0.0094 50  
D 0.0592 0.0604 0.0578 0.0027 0.0566 50  
U -0.0005 -0.0005 0.0001 -0.0006 0.0001 50  
Q 0.0598 0.0610 0.0577 0.0032 0.0565 50  
B 0.1216 0.1213 0.1219 -0.0006 0.1223 50  
g 0.6892 0.7047 0.6876 0.0172 0.6720 50  
gp 0.0917 0.0927 0.0910 0.0018 0.0899 50

## Nomogram for mod\_cc

plot(nomogram(mod\_cc, fun = plogis, funlabel = "Pr(10-year CHD)"))



# Using the Singly Imputed Data to fit the 4-predictor Model

## Fit mod\_si after single imputation

d <- datadist(fram\_si)  
options(datadist = "d")  
  
mod\_si <- lrm(chd10 ~ glucose + smoker + sbp + educ,  
 data = fram\_si, x = TRUE, y = TRUE)  
  
mod\_si

Logistic Regression Model  
  
lrm(formula = chd10 ~ glucose + smoker + sbp + educ, data = fram\_si,   
 x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 4238 LR chi2 237.48 R2 0.095 C 0.676   
 0 3594 d.f. 6 R2(6,4238)0.053 Dxy 0.353   
 1 644 Pr(> chi2) <0.0001 R2(6,1638.4)0.132 gamma 0.353   
max |deriv| 3e-11 Brier 0.121 tau-a 0.091   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -5.5680 0.3063 -18.18 <0.0001   
glucose 0.0085 0.0016 5.38 <0.0001   
smoker 0.3166 0.0901 3.51 0.0004   
sbp 0.0232 0.0019 12.42 <0.0001   
educ=HS grad -0.4498 0.1095 -4.11 <0.0001   
educ=Some Coll -0.3044 0.1327 -2.29 0.0218   
educ=Coll grad -0.0755 0.1464 -0.52 0.6062

## Comparing the Coefficients (exponentiated)

* Comparing the slopes as odds ratios

round\_half\_up(exp(mod\_cc$coefficients),3)

Intercept glucose smoker sbp educ=HS grad   
 0.004 1.008 1.367 1.024 0.627   
educ=Some Coll educ=Coll grad   
 0.675 0.873

round\_half\_up(exp(mod\_si$coefficients),3)

Intercept glucose smoker sbp educ=HS grad   
 0.004 1.009 1.372 1.023 0.638   
educ=Some Coll educ=Coll grad   
 0.738 0.927

## Comparing Model Summaries

| Summary | mod\_si | mod\_cc |
| --- | --- | --- |
| Obs | 4238 | 3753 |
| 0 | 3594 | 3174 |
| 1 | 644 | 579 |
| Nagelkerke | 0.095 | 0.100 |
| Brier Score | 0.121 | 0.122 |
| C | 0.676 | 0.682 |
| Dxy | 0.353 | 0.363 |

## Validate mod\_si Summary Statistics

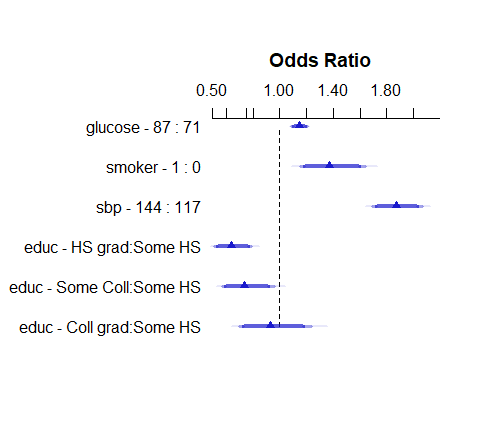
set.seed(4322); validate(mod\_si, B = 50)

index.orig training test optimism index.corrected n  
Dxy 0.3529 0.3488 0.3486 0.0002 0.3527 50  
R2 0.0950 0.0941 0.0929 0.0013 0.0938 50  
Intercept 0.0000 0.0000 -0.0157 0.0157 -0.0157 50  
Slope 1.0000 1.0000 0.9898 0.0102 0.9898 50  
Emax 0.0000 0.0000 0.0052 0.0052 0.0052 50  
D 0.0558 0.0552 0.0545 0.0008 0.0550 50  
U -0.0005 -0.0005 0.0001 -0.0005 0.0001 50  
Q 0.0563 0.0557 0.0544 0.0013 0.0550 50  
B 0.1206 0.1205 0.1209 -0.0004 0.1210 50  
g 0.6700 0.6684 0.6618 0.0066 0.6634 50  
gp 0.0882 0.0877 0.0871 0.0005 0.0877 50

* Since , validated C = 0.5 + (.3527/2) = 0.676

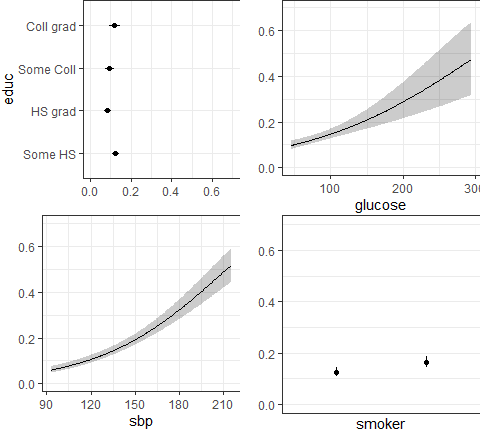
## Plot of Effects using mod\_si

plot(summary(mod\_si))



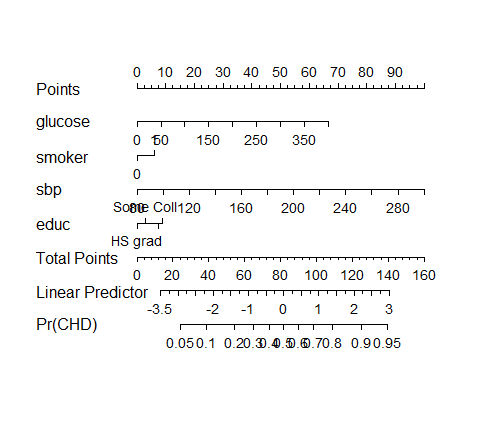
## Predict results for mod\_si

ggplot(Predict(mod\_si, fun = plogis))



## Nomogram for mod\_si

plot(nomogram(mod\_si, fun = plogis,  
 fun.at = c(0.05, seq(0.1, 0.9, by = 0.1), 0.95),  
 funlabel = "Pr(CHD)"))



* fun.at used to show us specific Pr(CHD) cutpoints

## Fit with glm() instead?

mod\_si\_glm <- glm(chd10 ~ glucose + smoker + sbp + educ,  
 data = fram\_si, family = binomial(link = "logit"))  
  
mod\_si\_glm

Call: glm(formula = chd10 ~ glucose + smoker + sbp + educ, family = binomial(link = "logit"),   
 data = fram\_si)  
  
Coefficients:  
 (Intercept) glucose smoker sbp educHS grad   
 -5.567956 0.008491 0.316589 0.023163 -0.449817   
educSome Coll educColl grad   
 -0.304367 -0.075459   
  
Degrees of Freedom: 4237 Total (i.e. Null); 4231 Residual  
Null Deviance: 3612   
Residual Deviance: 3374 AIC: 3388

## glance and tidy for mod\_si\_glm

glance(mod\_si\_glm)

# A tibble: 1 × 8  
 null.deviance df.null logLik AIC BIC deviance df.residual nobs  
 <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <int>  
1 3612. 4237 -1687. 3388. 3433. 3374. 4231 4238

tidy(mod\_si\_glm, conf.int = TRUE, conf.level = 0.90)

# A tibble: 7 × 7  
 term estimate std.error statistic p.value conf.low conf.high  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) -5.57 0.306 -18.2 7.28e-74 -6.08 -5.07   
2 glucose 0.00849 0.00158 5.38 7.54e- 8 0.00591 0.0111  
3 smoker 0.317 0.0901 3.51 4.40e- 4 0.169 0.465   
4 sbp 0.0232 0.00187 12.4 2.04e-35 0.0201 0.0262  
5 educHS grad -0.450 0.109 -4.11 3.96e- 5 -0.631 -0.271   
6 educSome Coll -0.304 0.133 -2.29 2.18e- 2 -0.526 -0.0888  
7 educColl grad -0.0755 0.146 -0.515 6.06e- 1 -0.320 0.162

## Confusion Matrix for mod\_si\_glm

mod\_si\_aug <- augment(mod\_si\_glm, type.predict = "response")  
  
cm\_si <- confusionMatrix(  
 data = factor(mod\_si\_aug$.fitted >= 0.5),  
 reference = factor(mod\_si\_aug$chd10 == 1),  
 positive = "TRUE")  
  
cm\_si

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 3574 618  
 TRUE 20 26  
   
 Accuracy : 0.8495   
 95% CI : (0.8383, 0.8601)  
 No Information Rate : 0.848   
 P-Value [Acc > NIR] : 0.4088   
   
 Kappa : 0.0562   
   
 Mcnemar's Test P-Value : <2e-16   
   
 Sensitivity : 0.040373   
 Specificity : 0.994435   
 Pos Pred Value : 0.565217   
 Neg Pred Value : 0.852576   
 Prevalence : 0.151958   
 Detection Rate : 0.006135   
 Detection Prevalence : 0.010854   
 Balanced Accuracy : 0.517404   
   
 'Positive' Class : TRUE

## Maximize Sensitivity + Specificity?

cp <- cutpointr(data = mod\_si\_aug, .fitted, chd10,   
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class is 1

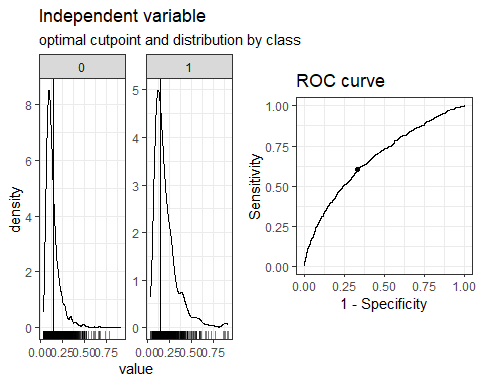
Assuming the positive class has higher x values

summary(cp)

Method: maximize\_metric   
Predictor: .fitted   
Outcome: chd10   
Direction: >=   
  
 AUC n n\_pos n\_neg  
 0.6765 4238 644 3594  
  
 optimal\_cutpoint sum\_sens\_spec acc sensitivity specificity tp fn fp tn  
 0.1485 1.2715 0.66 0.6009 0.6706 387 257 1184 2410  
  
Predictor summary:   
 Data Min. 5% 1st Qu. Median Mean 3rd Qu.  
 Overall 0.03677834 0.06336337 0.09460486 0.1260609 0.1519585 0.1782240  
 0 0.03677834 0.06136952 0.09203277 0.1211495 0.1420359 0.1666424  
 1 0.03797389 0.07710591 0.11673629 0.1696641 0.2073339 0.2521816  
 95% Max. SD NAs  
 0.3312963 0.9275036 0.09252701 0  
 0.2897352 0.7934734 0.07900552 0  
 0.4584220 0.9275036 0.13384091 0

## Plotting the cutpointr results

plot(cp)



## Confusion Matrix for mod\_si\_glm

* “Optimized” Rule: Predict CHD = 1 if .fitted .1485

mod\_si\_aug <- augment(mod\_si\_glm, type.predict = "response")  
  
cm\_si\_opt <- confusionMatrix(  
 data = factor(mod\_si\_aug$.fitted >= 0.1485),  
 reference = factor(mod\_si\_aug$chd10 == 1),  
 positive = "TRUE")  
  
cm\_si\_opt

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 2410 257  
 TRUE 1184 387  
   
 Accuracy : 0.66   
 95% CI : (0.6455, 0.6742)  
 No Information Rate : 0.848   
 P-Value [Acc > NIR] : 1   
   
 Kappa : 0.1707   
   
 Mcnemar's Test P-Value : <2e-16   
   
 Sensitivity : 0.60093   
 Specificity : 0.67056   
 Pos Pred Value : 0.24634   
 Neg Pred Value : 0.90364   
 Prevalence : 0.15196   
 Detection Rate : 0.09132   
 Detection Prevalence : 0.37069   
 Balanced Accuracy : 0.63575   
   
 'Positive' Class : TRUE

# Using Multiple Imputation: The 4-predictor Model

## Fit the Imputation Model first

We’ll use aregImpute here, and create 20 imputed sets.

* These imputations use only the 6 variables in our chd\_10 models.

set.seed(432123)  
dd <- datadist(fram\_start)  
options(datadist = "dd")  
  
fit\_imp <-   
 aregImpute(~ chd10 + glucose + smoker + sbp + educ,   
 nk = c(0, 3:5), tlinear = FALSE, data = fram\_start,  
 B = 10, n.impute = 20, pr = FALSE)

* fram\_start includes just our 6 variables (plus subj\_id) and includes missing glucose and educ.

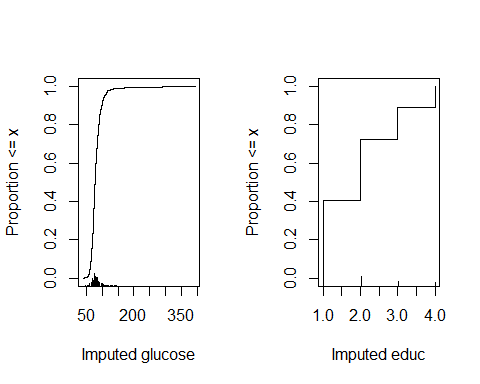
## Imputation Results

fit\_imp

Multiple Imputation using Bootstrap and PMM  
  
aregImpute(formula = ~chd10 + glucose + smoker + sbp + educ,   
 data = fram\_start, n.impute = 20, nk = c(0, 3:5), tlinear = FALSE,   
 pr = FALSE, B = 10)  
  
n: 4238 p: 5 Imputations: 20 nk: 0   
  
Number of NAs:  
 chd10 glucose smoker sbp educ   
 0 388 0 0 105   
  
 type d.f.  
chd10 l 1  
glucose s 1  
smoker l 1  
sbp s 1  
educ c 3  
  
R-squares for Predicting Non-Missing Values for Each Variable  
Using Last Imputations of Predictors  
glucose educ   
 0.036 0.027   
  
Resampling results for determining the complexity of imputation models  
  
Variable being imputed: glucose   
 nk=0 nk=3 nk=4 nk=5  
Bootstrap bias-corrected R^2 0.0289 0.0213 0.0302 0.0253  
10-fold cross-validated R^2 0.0315 0.0308 0.0298 0.0331  
Bootstrap bias-corrected mean |error| 12.4357 24.1861 19.8817 20.8650  
10-fold cross-validated mean |error| 81.7660 24.3909 22.0712 21.6855  
Bootstrap bias-corrected median |error| 8.5662 19.4253 13.9289 14.6123  
10-fold cross-validated median |error| 77.9625 20.0025 16.6750 15.6701  
  
Variable being imputed: educ   
 nk=0 nk=3 nk=4 nk=5  
Bootstrap bias-corrected R^2 0.0194 0.0142 0.0194 0.0196  
10-fold cross-validated R^2 0.0212 0.0230 0.0181 0.0191  
Bootstrap bias-corrected mean |error| 0.9795 0.9837 0.9833 0.9819  
10-fold cross-validated mean |error| 1.0298 1.0490 1.0278 1.0934  
Bootstrap bias-corrected median |error| 1.0000 1.0000 1.0000 1.0000  
10-fold cross-validated median |error| 1.0000 1.0000 1.0000 1.0000

## Multiply Imputed Values

par(mfrow=c(1,2)); plot(fit\_imp); par(mfrow = c(1,1))



## Needs for multiple imputation

* Appropriate datadist including missing values (fram\_start)
* Imputation Model

fit\_imp <-   
 aregImpute(~ chd10 + glucose + smoker + sbp + educ,   
 nk = c(0, 3:5), tlinear = FALSE, data = fram\_orig,  
 B = 10, n.impute = 20, pr = FALSE)

* Outcome Model will be of the following form, based on mod\_cc…

lrm(chd10 ~ glucose + smoker + sbp + educ, x = TRUE, y = TRUE)

## Fitting mod\_mi

mod\_mi <-   
 fit.mult.impute(chd10 ~ glucose + smoker + sbp + educ,  
 fitter = lrm, xtrans = fit\_imp,   
 data = fram\_start,   
 fitargs = list(x = TRUE, y = TRUE), pr = FALSE)

* data = fram\_start (which includes NA values)
* xtrans = fit\_imp (results from multiple imputation)
* fitter = lrm (we could actually use glm too, with different fitargs)
* pr = FALSE avoids a long printout we don’t need

## Model mod\_mi (using 20 imps.)

mod\_mi

Logistic Regression Model  
  
fit.mult.impute(formula = chd10 ~ glucose + smoker + sbp + educ,   
 fitter = lrm, xtrans = fit\_imp, data = fram\_start, pr = FALSE,   
 fitargs = list(x = TRUE, y = TRUE))  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 4238 LR chi2 236.68 R2 0.095 C 0.677   
 0 3594 d.f. 6 R2(6,4238)0.053 Dxy 0.353   
 1 644 Pr(> chi2) <0.0001 R2(6,1638.4)0.131 gamma 0.353   
max |deriv| 2e-11 Brier 0.121 tau-a 0.091   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -5.5450 0.3107 -17.85 <0.0001   
glucose 0.0082 0.0017 4.85 <0.0001   
smoker 0.3175 0.0902 3.52 0.0004   
sbp 0.0232 0.0019 12.41 <0.0001   
educ=HS grad -0.4517 0.1107 -4.08 <0.0001   
educ=Some Coll -0.3008 0.1351 -2.23 0.0260   
educ=Coll grad -0.0861 0.1481 -0.58 0.5610

## Comparing the Coefficients (exponentiated)

* I’ll just compare the two models using imputation…

round\_half\_up(exp(mod\_mi$coefficients),3)

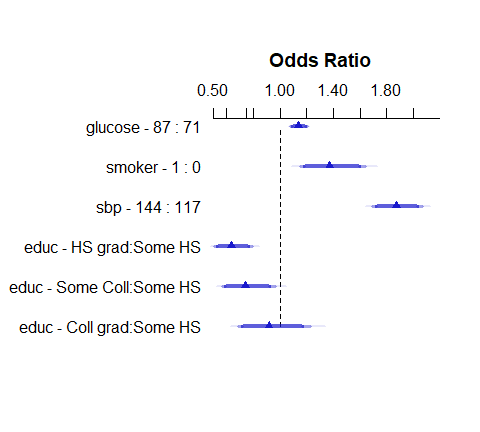
Intercept glucose smoker sbp educ=HS grad   
 0.004 1.008 1.374 1.023 0.637   
educ=Some Coll educ=Coll grad   
 0.740 0.918

round\_half\_up(exp(mod\_si$coefficients),3)

Intercept glucose smoker sbp educ=HS grad   
 0.004 1.009 1.372 1.023 0.638   
educ=Some Coll educ=Coll grad   
 0.738 0.927

## Plot of Effects using mod\_mi

plot(summary(mod\_mi))



## Summaries Comparing 3 Approaches

| Summary | mod\_mi | mod\_si | mod\_cc |
| --- | --- | --- | --- |
| Obs | 4238 | 4238 | 3753 |
| 0 | 3594 | 3594 | 3174 |
| 1 | 644 | 644 | 579 |
| Nagelkerke | 0.095 | 0.095 | 0.100 |
| Brier Score | 0.121 | 0.121 | 0.122 |
| C | 0.677 | 0.676 | 0.682 |
| Dxy | 0.353 | 0.353 | 0.363 |

* What might cause these to look meaningfully different?

## Validate mod\_mi Summary Statistics

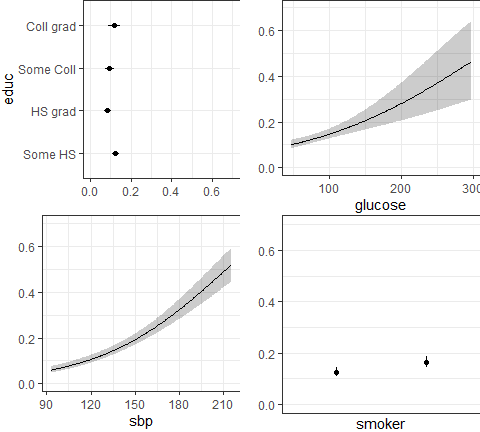
set.seed(4323)  
validate(mod\_mi, B = 50)

index.orig training test optimism index.corrected n  
Dxy 0.3568 0.3565 0.3531 0.0035 0.3533 50  
R2 0.0947 0.0983 0.0952 0.0030 0.0917 50  
Intercept 0.0000 0.0000 -0.0234 0.0234 -0.0234 50  
Slope 1.0000 1.0000 0.9911 0.0089 0.9911 50  
Emax 0.0000 0.0000 0.0066 0.0066 0.0066 50  
D 0.0556 0.0580 0.0559 0.0021 0.0535 50  
U -0.0005 -0.0005 0.0001 -0.0006 0.0001 50  
Q 0.0561 0.0585 0.0558 0.0026 0.0534 50  
B 0.1205 0.1211 0.1206 0.0005 0.1200 50  
g 0.6692 0.6815 0.6724 0.0091 0.6601 50  
gp 0.0882 0.0900 0.0883 0.0017 0.0864 50

* Optimism-corrected C = 0.5 + (0.3533/2) = 0.677

## Predict results for mod\_mi

ggplot(Predict(mod\_mi, fun = plogis))



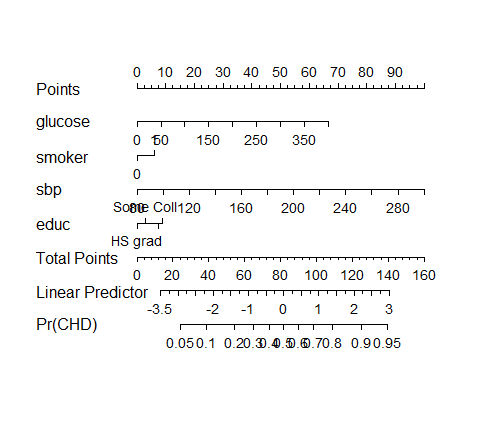
## Is collinearity a problem?

rms::vif(mod\_mi)

glucose smoker sbp educ=HS grad educ=Some Coll   
 1.012944 1.034353 1.047758 1.137513 1.119299   
educ=Coll grad   
 1.110563

## Nomogram for mod\_mi

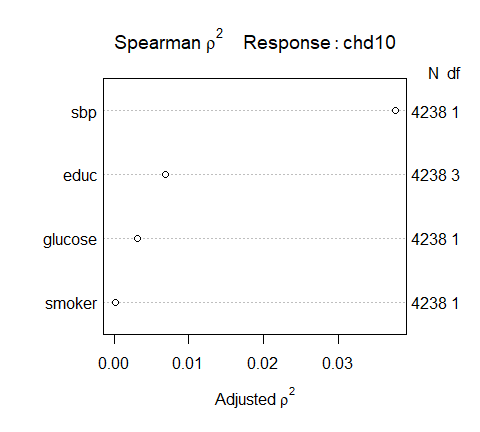
plot(nomogram(mod\_si, fun = plogis,  
 fun.at = c(0.05, seq(0.1, 0.9, by = 0.1), 0.95),  
 funlabel = "Pr(CHD)"))



# Considering Non-Linear Terms

## Spearman Plot (using fram\_si)

plot(spearman2(chd10 ~ glucose + smoker + sbp + educ, data = fram\_si))



## Adding some non-linear terms

* We’ll add a restricted cubic spline with 5 knots in sbp
* and an interaction between the educ factor and the linear effect of sbp,
* and a quadratic polynomial in glucose

to our main effects model, just to show how to do them…

* I’ll just show the results including the multiple imputation, since if you can get those, you should have little difficulty instead applying the single imputation or the complete case analysis.

## mod\_big using 20 imputations

* mod\_big incorporates our non-linear terms.

mod\_big <-   
 fit.mult.impute(  
 chd10 ~ rcs(sbp, 5) + pol(glucose, 2) +   
 smoker + educ + educ %ia% sbp,  
 fitter = lrm, xtrans = fit\_imp,   
 data = fram\_start, fitargs = list(x = TRUE, y = TRUE),  
 pr = FALSE)

## Results of mod\_big

mod\_big

Logistic Regression Model  
  
fit.mult.impute(formula = chd10 ~ rcs(sbp, 5) + pol(glucose,   
 2) + smoker + educ + educ %ia% sbp, fitter = lrm, xtrans = fit\_imp,   
 data = fram\_start, pr = FALSE, fitargs = list(x = TRUE, y = TRUE))  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 4238 LR chi2 243.86 R2 0.097 C 0.678   
 0 3594 d.f. 13 R2(13,4238)0.053 Dxy 0.357   
 1 644 Pr(> chi2) <0.0001 R2(13,1638.4)0.131 gamma 0.357   
max |deriv| 0.01 Brier 0.120 tau-a 0.092   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -3.2433 2.1149 -1.53 0.1251   
sbp 0.0033 0.0190 0.17 0.8635   
sbp' 0.1772 0.1837 0.96 0.3349   
sbp'' -0.5110 0.6403 -0.80 0.4248   
sbp''' 0.3704 0.6493 0.57 0.5684   
glucose 0.0061 0.0052 1.17 0.2438   
glucose^2 0.0000 0.0000 0.44 0.6622   
smoker 0.3205 0.0903 3.55 0.0004   
educ=HS grad -0.4241 0.6397 -0.66 0.5073   
educ=Some Coll -1.4303 0.8104 -1.76 0.0776   
educ=Coll grad -1.0843 0.9401 -1.15 0.2487   
educ=HS grad \* sbp -0.0003 0.0045 -0.06 0.9548   
educ=Some Coll \* sbp 0.0082 0.0057 1.43 0.1541   
educ=Coll grad \* sbp 0.0073 0.0068 1.08 0.2799

## mod\_big with robust sandwich variance estimates

Here we add robust = TRUE to get robust sandwich variance estimates into Rubin’s rule for combining our imputations.

mod\_bigr <-   
 fit.mult.impute(  
 chd10 ~ rcs(sbp, 5) + pol(glucose, 2) +   
 smoker + educ + educ %ia% sbp,  
 fitter = lrm, xtrans = fit\_imp, robust = TRUE,  
 data = fram\_start, fitargs = list(x = TRUE, y = TRUE),  
 pr = FALSE)

## Results using Robust SEs

mod\_bigr

Logistic Regression Model  
  
fit.mult.impute(formula = chd10 ~ rcs(sbp, 5) + pol(glucose,   
 2) + smoker + educ + educ %ia% sbp, fitter = lrm, xtrans = fit\_imp,   
 data = fram\_start, robust = TRUE, pr = FALSE, fitargs = list(x = TRUE,   
 y = TRUE))  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 4238 LR chi2 243.86 R2 0.097 C 0.678   
 0 3594 d.f. 13 R2(13,4238)0.053 Dxy 0.357   
 1 644 Pr(> chi2) <0.0001 R2(13,1638.4)0.131 gamma 0.357   
max |deriv| 0.01 Brier 0.120 tau-a 0.092   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -3.2433 2.3538 -1.38 0.1682   
sbp 0.0033 0.0211 0.15 0.8773   
sbp' 0.1772 0.2000 0.89 0.3758   
sbp'' -0.5110 0.6849 -0.75 0.4556   
sbp''' 0.3704 0.6805 0.54 0.5863   
glucose 0.0061 0.0057 1.07 0.2853   
glucose^2 0.0000 0.0000 0.38 0.7013   
smoker 0.3205 0.0901 3.56 0.0004   
educ=HS grad -0.4241 0.6359 -0.67 0.5048   
educ=Some Coll -1.4303 0.8061 -1.77 0.0760   
educ=Coll grad -1.0843 0.9791 -1.11 0.2681   
educ=HS grad \* sbp -0.0003 0.0045 -0.06 0.9544   
educ=Some Coll \* sbp 0.0082 0.0057 1.43 0.1525   
educ=Coll grad \* sbp 0.0073 0.0071 1.03 0.3017

## Impact of Robust Estimates

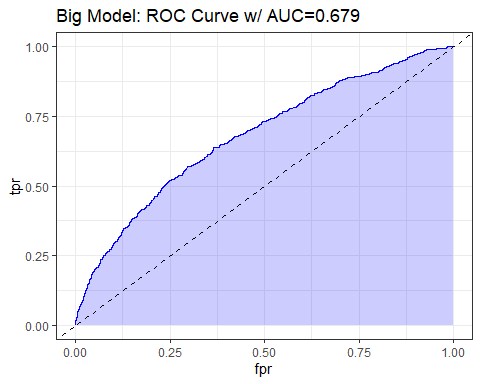
* No changes to anything above the coefficients (Likelihood Ratio Test, Discrimination or Rank Discrimination Indexes)

mod\_big mod\_big\_r  
 Coef S.E. Pr(>|Z|) Coef S.E. Pr(>|Z|)  
Intercept -3.2433 2.1149 0.1251 -3.2433 2.3538 0.1682   
sbp 0.0033 0.0190 0.8635 0.0033 0.0211 0.8773  
sbp' 0.1772 0.1837 0.3349 0.1772 0.2000 0.3758  
sbp'' -0.5110 0.6403 0.4248 -0.5110 0.6849 0.4556  
sbp''' 0.3704 0.6493 0.5684 0.3704 0.6805 0.5863  
glucose 0.0061 0.0052 0.2438 0.0061 0.0057 0.2853  
glucose^2 0.0000 0.0000 0.6622 0.0000 0.0000 0.7013  
smoker 0.3205 0.0903 0.0004 0.3205 0.0901 0.0004  
educ=HS grad -0.4241 0.6397 0.5073 -0.4241 0.6359 0.5048  
educ=Some Coll -1.4303 0.8104 0.0776 -1.4303 0.8061 0.0760  
educ=Coll grad -1.0843 0.9401 0.2487 -1.0843 0.9791 0.2681  
educ=HS grad \* sbp -0.0003 0.0045 0.9548 -0.0003 0.0045 0.9544  
educ=Some Coll \* sbp 0.0082 0.0057 0.1541 0.0082 0.0057 0.1525  
educ=Coll grad \* sbp 0.0073 0.0068 0.2799 0.0073 0.0071 0.3017

## mod\_big vs. mod\_mi comparison

| Summary | mod\_big | mod\_mi |
| --- | --- | --- |
| Obs | 4238 | 4238 |
| 0 | 3594 | 3594 |
| 1 | 644 | 644 |
| Nagelkerke | 0.097 | 0.095 |
| Brier Score | 0.120 | 0.121 |
| C | 0.678 | 0.677 |
| Dxy | 0.357 | 0.353 |

## ROC Curve for mod\_big



## ANOVA for the big fit?

* How many df did we add in non-linear + interaction terms?

anova(mod\_big)

Wald Statistics Response: chd10   
  
 Factor Chi-Square d.f. P   
 sbp (Factor+Higher Order Factors) 160.32 7 <.0001  
 All Interactions 3.09 3 0.3780  
 Nonlinear 3.04 3 0.3857  
 glucose 23.18 2 <.0001  
 Nonlinear 0.19 1 0.6622  
 smoker 12.61 1 0.0004  
 educ (Factor+Higher Order Factors) 21.45 6 0.0015  
 All Interactions 3.09 3 0.3780  
 educ \* sbp (Factor+Higher Order Factors) 3.09 3 0.3780  
 TOTAL NONLINEAR 3.18 4 0.5287  
 TOTAL NONLINEAR + INTERACTION 6.91 7 0.4380  
 TOTAL 219.67 13 <.0001

## Is collinearity involved now?

rms::vif(mod\_big)

sbp sbp' sbp''   
 107.677097 10882.179174 28610.682387   
 sbp''' glucose glucose^2   
 5779.211307 9.392617 9.360416   
 smoker educ=HS grad educ=Some Coll   
 1.036394 38.187396 39.301981   
 educ=Coll grad educ=HS grad \* sbp educ=Some Coll \* sbp   
 43.847912 37.856282 38.950457   
educ=Coll grad \* sbp   
 43.321083

## Validate mod\_big Summary Statistics

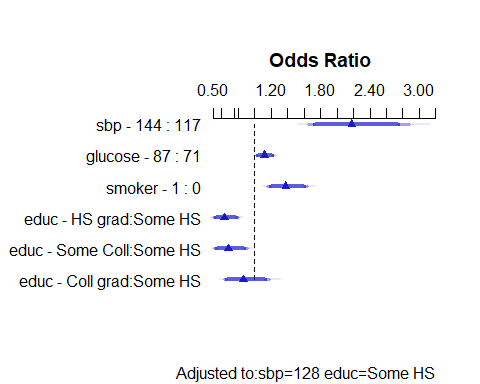
set.seed(4324); validate(mod\_big, B = 50)

index.orig training test optimism index.corrected n  
Dxy 0.3597 0.3692 0.3536 0.0156 0.3442 50  
R2 0.0975 0.1064 0.0951 0.0113 0.0862 50  
Intercept 0.0000 0.0000 -0.0845 0.0845 -0.0845 50  
Slope 1.0000 1.0000 0.9449 0.0551 0.9449 50  
Emax 0.0000 0.0000 0.0283 0.0283 0.0283 50  
D 0.0573 0.0627 0.0558 0.0069 0.0504 50  
U -0.0005 -0.0005 0.0001 -0.0005 0.0001 50  
Q 0.0578 0.0632 0.0558 0.0074 0.0504 50  
B 0.1202 0.1194 0.1206 -0.0012 0.1214 50  
g 0.7092 0.7439 0.7002 0.0436 0.6655 50  
gp 0.0917 0.0949 0.0899 0.0050 0.0867 50

* Optimism-Corrected C = 0.5 + (.3442/2) = .672

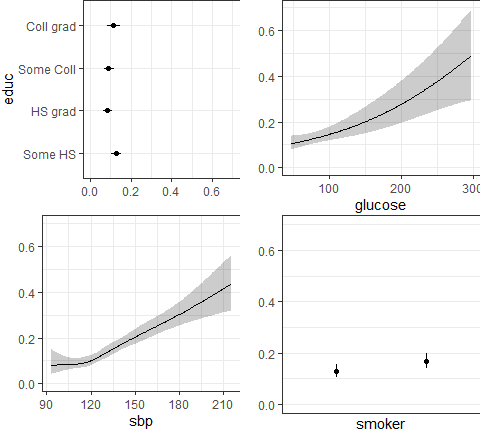
## Plot of Effects using mod\_big

plot(summary(mod\_big))



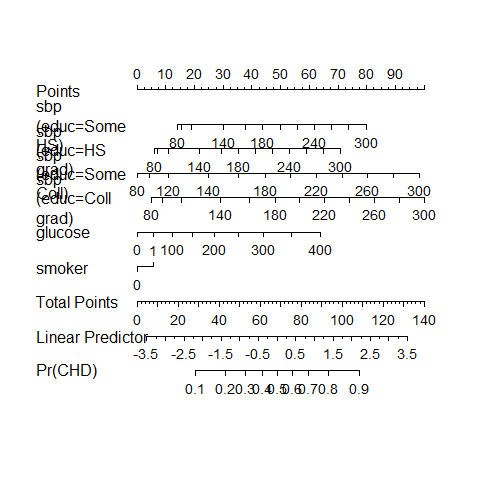
## Predict results for mod\_big

ggplot(Predict(mod\_big, fun = plogis))



## Nomogram for mod\_big

plot(nomogram(mod\_big, fun = plogis, funlabel = "Pr(CHD)"))



## glm() fit with aregImpute()?

mod\_big\_glm <-   
 fit.mult.impute(  
 chd10 ~ rcs(sbp, 5) + pol(glucose, 2) +   
 smoker + educ + educ %ia% sbp,  
 fitter = glm, xtrans = fit\_imp,   
 data = fram\_start,   
 fitargs = list(family = binomial(link = "logit")),  
 pr = FALSE)

## Results for mod\_big\_glm

mod\_big\_glm

Call: fit.mult.impute(formula = chd10 ~ rcs(sbp, 5) + pol(glucose,   
 2) + smoker + educ + educ %ia% sbp, fitter = glm, xtrans = fit\_imp,   
 data = fram\_start, pr = FALSE, fitargs = list(family = binomial(link = "logit")))  
  
Coefficients:  
 (Intercept) rcs(sbp, 5)sbp   
 -3.243e+00 3.261e-03   
 rcs(sbp, 5)sbp' rcs(sbp, 5)sbp''   
 1.772e-01 -5.110e-01   
 rcs(sbp, 5)sbp''' pol(glucose, 2)glucose   
 3.704e-01 6.083e-03   
 pol(glucose, 2)glucose^2 smoker   
 6.750e-06 3.205e-01   
 educHS grad educSome Coll   
 -4.241e-01 -1.430e+00   
 educColl grad educ %ia% sbpeduc=HS grad \* sbp   
 -1.084e+00 -2.556e-04   
educ %ia% sbpeduc=Some Coll \* sbp educ %ia% sbpeduc=Coll grad \* sbp   
 8.169e-03 7.321e-03   
  
Degrees of Freedom: 4237 Total (i.e. Null); 4224 Residual  
Null Deviance: 3612   
Residual Deviance: 3362 AIC: 3390

## glance and tidy results

glance(mod\_big\_glm)

Warning: The `glance()` method for objects of class `fit.mult.impute` is not maintained by the broom team, and is only supported through the `glm` tidier method. Please be cautious in interpreting and reporting broom output.  
  
This warning is displayed once per session.

# A tibble: 1 × 8  
 null.deviance df.null logLik AIC BIC deviance df.residual nobs  
 <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <int>  
1 3612. 4237 -1681. 3390. 3479. 3362. 4224 4238

tidy(mod\_big\_glm, exponentiate = TRUE,   
 conf.int = TRUE, conf.level = 0.90)

# A tibble: 14 × 7  
 term estimate std.error statistic p.value conf.low conf.high  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 (Intercept) 0.0390 2.11 -1.54 1.25e-1 0.000952 1.00   
 2 rcs(sbp, 5)sbp 1.00 0.0190 0.172 8.63e-1 0.974 1.04   
 3 rcs(sbp, 5)sbp' 1.19 0.184 0.964 3.35e-1 0.872 1.60   
 4 rcs(sbp, 5)sbp'' 0.600 0.640 -0.798 4.25e-1 0.217 1.78   
 5 rcs(sbp, 5)sbp''' 1.45 0.649 0.570 5.68e-1 0.481 4.08   
 6 pol(glucose, 2)gluco… 1.01 0.00511 1.19 2.34e-1 0.999 1.02   
 7 pol(glucose, 2)gluco… 1.00 0.0000152 0.444 6.57e-1 1.00 1.00   
 8 smoker 1.38 0.0903 3.55 3.84e-4 1.19 1.61   
 9 educHS grad 0.654 0.634 -0.669 5.03e-1 0.236 1.90   
10 educSome Coll 0.239 0.799 -1.79 7.36e-2 0.0562 0.782  
11 educColl grad 0.338 0.933 -1.16 2.45e-1 0.0634 1.37   
12 educ %ia% sbpeduc=HS… 1.00 0.00447 -0.0571 9.54e-1 0.992 1.01   
13 educ %ia% sbpeduc=So… 1.01 0.00565 1.45 1.48e-1 1.00 1.02   
14 educ %ia% sbpeduc=Co… 1.01 0.00673 1.09 2.77e-1 0.997 1.02

## ROC for mod\_big\_glm?

predict.prob1 <- predict(mod\_big\_glm, type = "response")  
roc1 <- pROC::roc(mod\_big\_glm$data$chd10, predict.prob1)

Setting levels: control = 0, case = 1

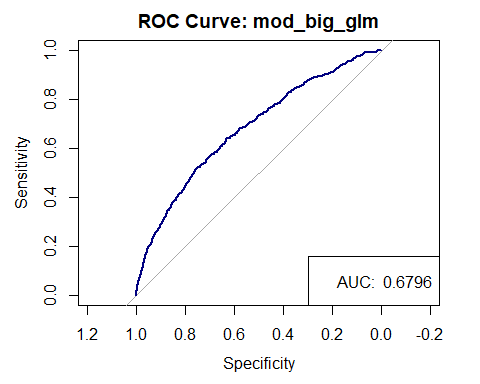
Setting direction: controls < cases

roc1

Call:  
roc.default(response = mod\_big\_glm$data$chd10, predictor = predict.prob1)  
  
Data: predict.prob1 in 3594 controls (mod\_big\_glm$data$chd10 0) < 644 cases (mod\_big\_glm$data$chd10 1).  
Area under the curve: 0.6796

## Plotting the ROC curve

plot(roc1, main = "ROC Curve: mod\_big\_glm", lwd = 2, col = "navy")  
legend('bottomright', legend = paste("AUC: ",   
 round\_half\_up(auc(roc1),4)))



## Confusion Matrix

mod\_big\_aug <- augment(mod\_big\_glm, type.predict = "response")  
  
cm2 <- confusionMatrix(  
 data = factor(mod\_big\_aug$.fitted >= 0.5),  
 reference = factor(mod\_big\_aug$chd10 == 1),  
 positive = "TRUE")  
  
cm2

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 3580 625  
 TRUE 14 19  
   
 Accuracy : 0.8492   
 95% CI : (0.8381, 0.8599)  
 No Information Rate : 0.848   
 P-Value [Acc > NIR] : 0.4255   
   
 Kappa : 0.0419   
   
 Mcnemar's Test P-Value : <2e-16   
   
 Sensitivity : 0.029503   
 Specificity : 0.996105   
 Pos Pred Value : 0.575758   
 Neg Pred Value : 0.851367   
 Prevalence : 0.151958   
 Detection Rate : 0.004483   
 Detection Prevalence : 0.007787   
 Balanced Accuracy : 0.512804   
   
 'Positive' Class : TRUE

## Maximize Sensitivity + Specificity

cp2 <- cutpointr(data = mod\_big\_aug, .fitted, chd10,   
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class is 1

Assuming the positive class has higher x values

summary(cp2)

Method: maximize\_metric   
Predictor: .fitted   
Outcome: chd10   
Direction: >=   
  
 AUC n n\_pos n\_neg  
 0.6799 4238 644 3594  
  
 optimal\_cutpoint sum\_sens\_spec acc sensitivity specificity tp fn fp  
 0.1434 1.2744 0.6354 0.6398 0.6347 412 232 1313  
 tn  
 2281  
  
Predictor summary:   
 Data Min. 5% 1st Qu. Median Mean 3rd Qu.  
 Overall 0.04252338 0.05987738 0.08832861 0.1250505 0.1518500 0.1880941  
 0 0.04252338 0.05863544 0.08544604 0.1199834 0.1417409 0.1752832  
 1 0.04876396 0.07255436 0.11637324 0.1792092 0.2082664 0.2663616  
 95% Max. SD NAs  
 0.3231884 0.9538913 0.09188942 0  
 0.2891345 0.8543735 0.07967312 0  
 0.4484090 0.9538913 0.12809846 0

## Confusion Matrix at .fitted >= .143

mod\_big\_aug <- augment(mod\_big\_glm, type.predict = "response")  
  
cm\_new <- confusionMatrix(  
 data = factor(mod\_big\_aug$.fitted >= 0.143),  
 reference = factor(mod\_big\_aug$chd10 == 1),  
 positive = "TRUE")  
  
cm\_new

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 2276 232  
 TRUE 1318 412  
   
 Accuracy : 0.6343   
 95% CI : (0.6196, 0.6488)  
 No Information Rate : 0.848   
 P-Value [Acc > NIR] : 1   
   
 Kappa : 0.1614   
   
 Mcnemar's Test P-Value : <2e-16   
   
 Sensitivity : 0.63975   
 Specificity : 0.63328   
 Pos Pred Value : 0.23815   
 Neg Pred Value : 0.90750   
 Prevalence : 0.15196   
 Detection Rate : 0.09722   
 Detection Prevalence : 0.40821   
 Balanced Accuracy : 0.63651   
   
 'Positive' Class : TRUE

## Next Time

Back to Linear Regression

* Variable (Feature) Selection in Linear Regression
* Ridge Regression and the Lasso