432 Class 10

https://thomaselove.github.io/432-2023/

2023-02-16

Today's Agenda

Fitting and evaluating logistic regression models with `lrm`

- The framingham example
  - Outcome: `chd10` = Developed coronary heart disease in next 10 years?
  - Creating "complete case" data: `fram_cc`
  - Single Imputation of Missing Values: `fram_sh`
- Use `lrm` to predict `chd10` using `glucose`, `smoker`, `sbp` and `educ`
  - on the complete cases (`fram_cc`)
  - accounting for missingness via single imputation (`fram_sh`)
  - 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(janitor)
library(knitr)
library(naniar)
library(simputation)
library(ROCR)
library(rms)
library(tidyverse)

theme_set(theme_bw())
```

Section 1

The “Framingham” Data

The Data

```r
fram_raw <- read_csv("c10/data/framingham.csv",
                     show_col_types = FALSE) |>
    clean_names()
```

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

- This particular data set, purportedly from the Framingham study, has been used by lots of people, in varied settings, with variations all over the net. I don't know who the originators were.

Data Cleanup

```
fram <- 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)
```

Data Descriptions (Main Variables Today)

The variables describe n = 4238 adults examined at baseline, then followed for 10 years to see if they developed incident coronary heart disease.

The main variables we'll use today in developing outcome models are:

Variable	Description
`subj_id`	identifying code added by Dr. Love
`chd10`	1 = coronary heart disease in next 10 years
`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 Descriptions (Other 11 variables)

Here are the other 11 variables in the `fram` data.

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 at time of exam
`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 kg/m^2
`hrate`	heart rate in beats per minute

Missing Data?

Our outcome `chd10` has no missing values.

```r
fram |> 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 the `fram` data are complete.
- The remaining 582 observations have something missing.

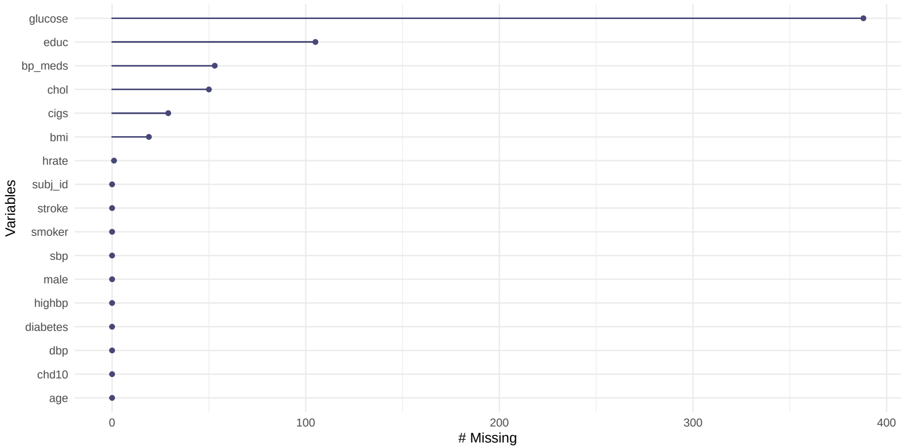
```r
n_case_complete(fram); pct_complete_case(fram)
```

```
[1] 3656
```

```
[1] 86.26711
```

Which variables are missing data?

```
gg_miss_var(fram)
```



Counts of Missing Data, by Variable

```r
miss_var_summary(fram) |>
    filter(n_miss > 0)
```

```
# A tibble: 7 x 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
```

Single Imputation

We will impute:

- 5 quantitative variables (glucose, bmi, cigs, chol and hrate)
- 1 binary variable (bp_meds), and
- 1 multi-categorical variable (educ)

```
fram_sh <- bind_shadow(fram)

fram_sh <- fram_sh |>
    data.frame() |>
    impute_pmm(bp_meds ~ highbp + sbp + dbp) |>
    impute_cart(educ ~ age + smoker + male) |>
    impute_pmm(cigs ~ smoker) |>
    impute_rlm(glucose + chol + hrate + bmi ~
                sbp + diabetes + age + highbp + stroke) |>
    tibble()
```

Check multi-categorical single imputation?

```
fram_sh |> count(educ_NA, educ)
```

```
# A tibble: 6 x 3
  educ_NA educ          n
  <fct>   <fct>     <int>
1 !NA     Some HS    1720
2 !NA     HS grad    1253
3 !NA     Some Coll   687
4 !NA     Coll grad   473
5 NA      Some HS      80
6 NA      HS grad      25
```

Do the values seem reasonable?

Data Sets for the rest of our work

```
fram_start <- fram |>
  select(subj_id, chd10, glucose, smoker, sbp, educ)

fram_cc <- fram_start |>
  drop_na()

fram_sh <- fram_sh  |>
  select(subj_id, chd10, glucose, smoker, sbp, educ,
         glucose_NA, educ_NA)
```

- `fram_start` includes all 4238 rows and the 6 columns we'll use, including 388 rows missing `glucose` and 105 missing `educ`.
- `fram_cc` includes only the 3753 complete rows on the 6 columns.
- `fram_sh` uses single imputation to get 4238 complete rows, on 8 columns, including the useful missingness indicators.

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_sh`)
3. Accounting for missingness via multiple imputation

Then, we'll consider adding several non-linear terms to the "four-predictor" models, and refit.

Section 2

Fitting a Four-Predictor Model using Complete Cases

A “Four Predictor” model

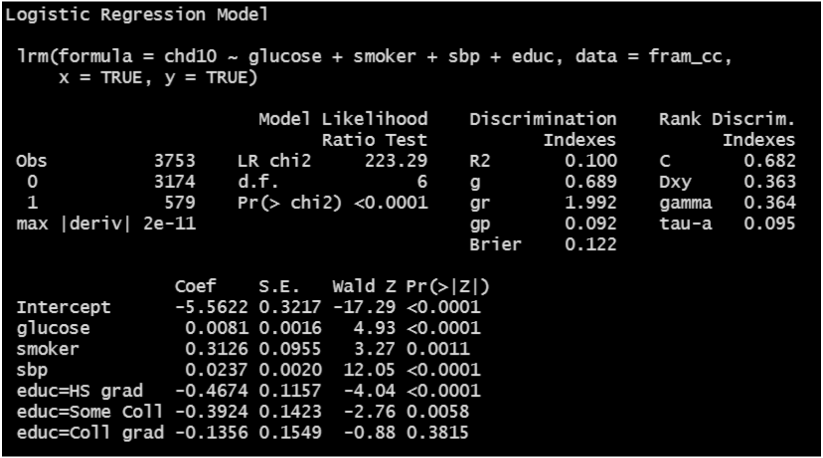
First, we’ll use the `fram_cc` data to perform a complete-case analysis and fix ideas.

```r
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...

```r
mod_cc <- lrm(outcome == "Yes" ~
                   glucose + smoker + sbp + educ,
            data = fram_cc, x = TRUE, y = TRUE)
```

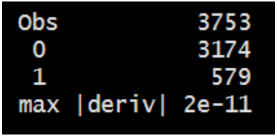
Main Output for `mod_cc`



```
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    g        0.689    Dxy     0.363
  1             579    Pr(> chi2) <0.0001    gr       1.992    gamma   0.364
 max |deriv| 2e-11                           gp       0.092    tau-a   0.095
                                             Brier    0.122

                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
```

- We'll walk through these summaries in the next few slides.
- Notes Section 21.2 provides additional details.

Deconstructing the `mod_cc` summaries (1/5)

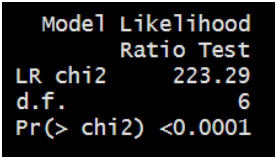


```
Obs          3753
 0           3174
 1            579
max |deriv| 2e-11
```

- `Obs` = The number of observations used to fit the model, with `0` = the number of zeros and `1` = the number of ones in our outcome, `chd10`.
- Also specified is the maximum absolute value of the derivative at the point where the maximum likelihood function was estimated.

All you're likely to care about is whether the iterative function-fitting process converged, and R will warn you in other ways if it doesn't.

Deconstructing the `mod_cc` summaries (2/5)



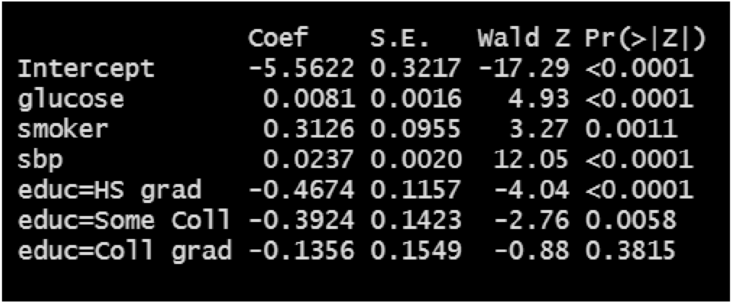
```
  Model Likelihood
        Ratio Test
LR chi2     223.29
d.f.             6
Pr(> chi2) <0.0001
```

- This is a global likelihood ratio test (drop in deviance test.)
- Likelihood Ratio χ^2 statistic $=$ null deviance - residual deviance
    - d.f. $=$ null degrees of freedom - residual degrees of freedom
- Pr($>$ chi2) is a *p* value obtained from comparison to a χ^2 distribution with appropriate d.f.

It's not saying much to suggest that some part of this logistic regression model has some detectable predictive value.

- The null hypothesis here (that the model has no predictive value at all) is rarely interesting in practical work.

Deconstructing the `mod_cc` summaries (3/4)

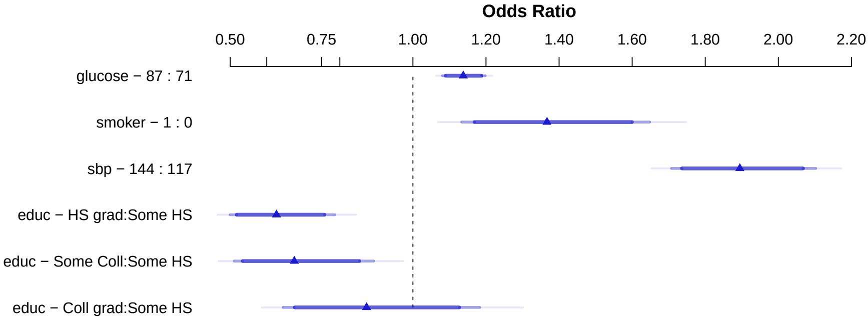


```
               Coef    S.E.   Wald Z Pr(>|Z|)
Intercept      -5.5622 0.3217 -17.29 <0.0001
glucose         0.0081 0.0016   4.93 <0.0001
smoker          0.3126 0.0955   3.27 0.0011
sbp             0.0237 0.0020  12.05 <0.0001
educ=HS grad   -0.4674 0.1157  -4.04 <0.0001
educ=Some Coll -0.3924 0.1423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

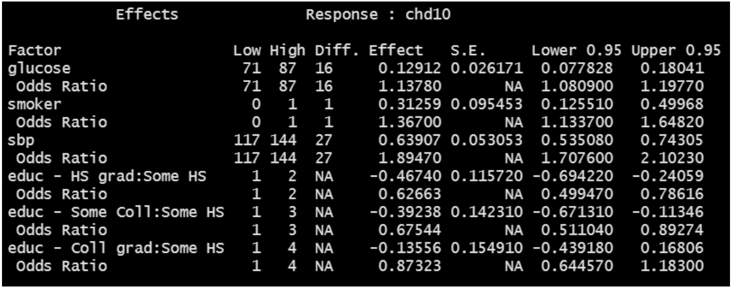
- How does each predictor appear to relate to 10-year risk?
  - Which is the baseline `educ` category?
  - Remember that these estimates are on the logit scale.
  - See the effect size discussion linked in today's README.

Plot of Effects using `mod_cc`

```
plot(summary(mod_cc))
```



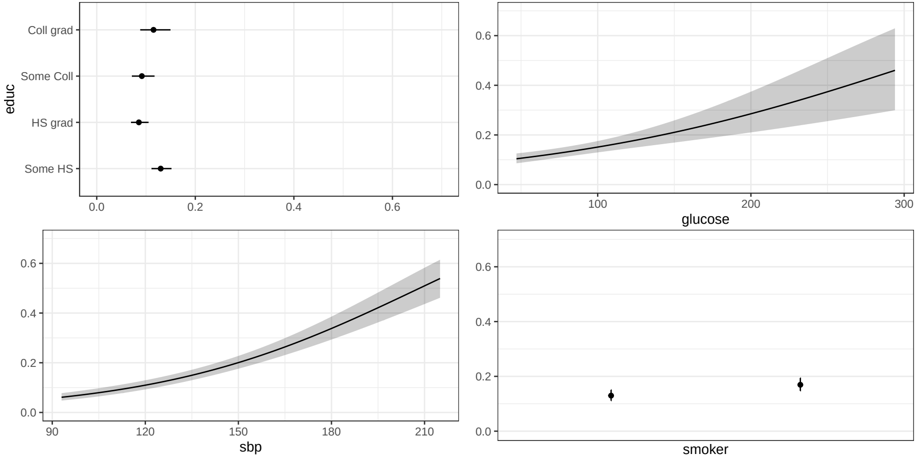
Effect Size Summary for `mod_cc`



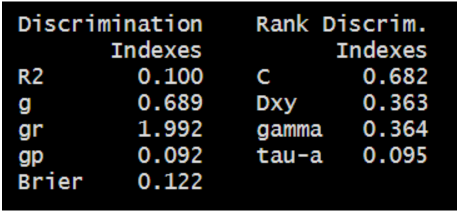
```
             Effects              Response : chd10

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

Predict results for `mod_cc`

```r
ggplot(Predict(mod_cc, fun = plogis))
```



Deconstructing the `mod_cc` summaries (4/4)



```
Discrimination    Rank Discrim.
       Indexes          Indexes
R2       0.100    C       0.682
g        0.689    Dxy     0.363
gr       1.992    gamma   0.364
gp       0.092    tau-a   0.095
Brier    0.122
```

The key indexes for our purposes are:

- Nagelkerke R^2, 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 R^2)

- In our model, Nagelkerke $R^2 = 0.100$

There are at least three ways to think about R^2 in linear regression, but when you move to a categorical outcome, not all of those ways can be expressed in the same statistic. See our Course Notes Section 21.2 for details.

The Nagelkerke R^2:

- 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).
- is 0 for the null model
- is larger (closer to 1) as the fitted model improves, although it's been criticized for being misleadingly high,
- AND a value of 0.100 no longer means 10% of anything.

A value of 0.100 indicates a model of pretty poor quality.

An Alternative: McFadden's R^2

Consider the McFadden R-square, which can be defined as 1 minus the ratio of (the model deviance over the deviance for the intercept-only model.)

To obtain this for our `mod_cc` run with `lrm`, we can use:

```r
1 - (mod_cc$deviance[2] / mod_cc$deviance[1])
```

```
[1] 0.069174
```

This McFadden R^2 corresponds well to the proportionate reduction in error interpretation of an R^2, but some people don't like it as well.

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're forecasting the probability P that it will rain on a given day.

- 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 = $(0.70 - 1)^2 = 0.09$.
- If the forecast is P = 0.3 and it rains, Brier = $(0.30 - 1)^2 = 0.49$.
- If the forecast is P = 0.5, the Brier score is $(0.50 - 1)^2 = 0.25$ regardless of whether it rains.

The Brier score can also be decomposed to assess calibration and discrimination separately.

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.

See Section 20.10 of our Course Notes for more details.

The C statistic (area under ROC curve) = 0.682

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

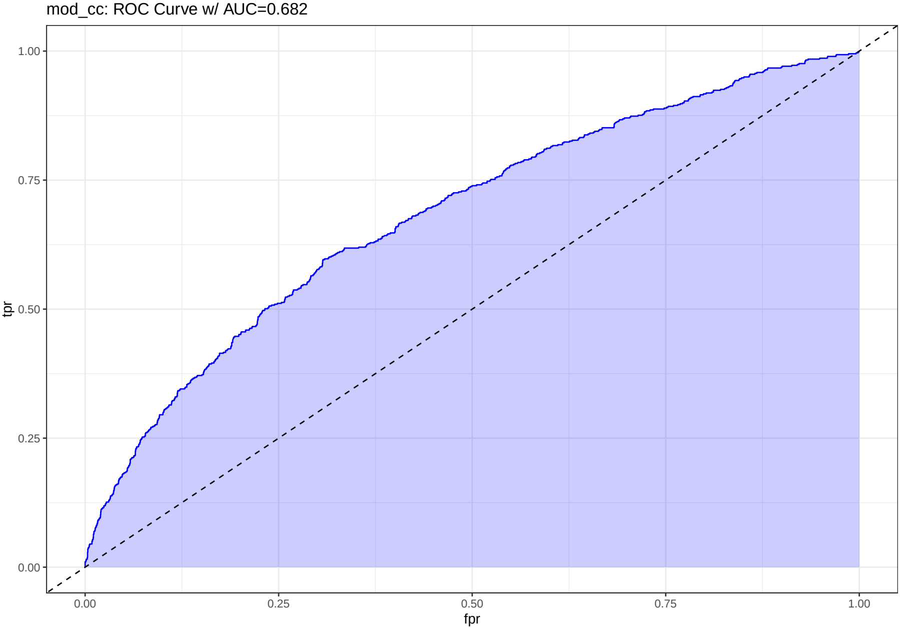
$$C = 0.5 + \frac{d}{2}, d = 2(C - .5)$$

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:
  - $C \geq 0.8$ usually indicates a moderately strong model (good discrimination)
  - $C \geq 0.9$ indicates a very strong model (excellent discrimination)

So 0.682 isn't good.

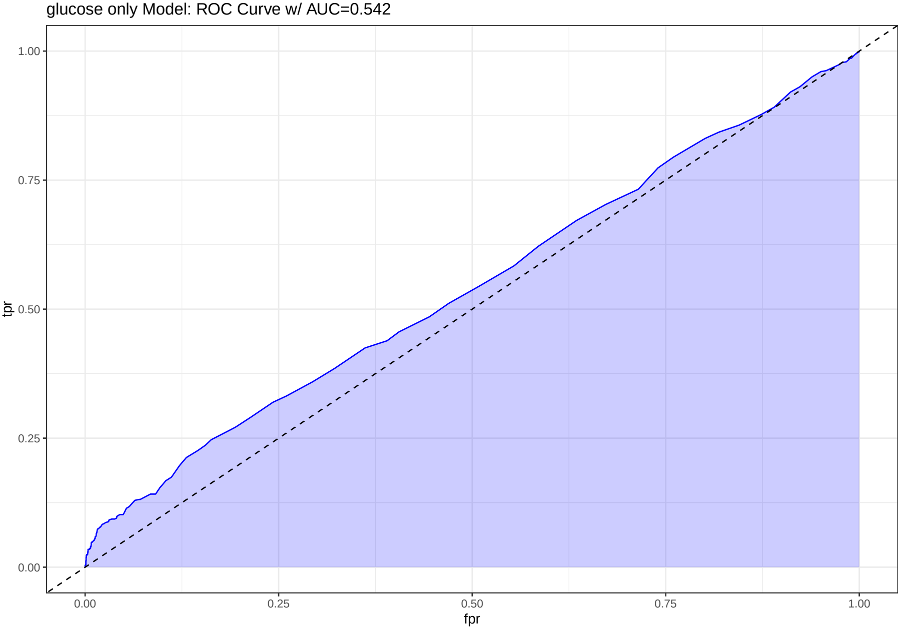
ROC Curve for our `mod_cc`



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))
```

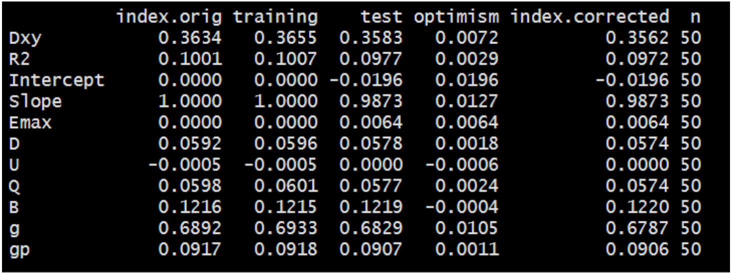
ROC Curve for a Simple Model (`glucose` only)



Validate Summary Statistics for `mod_cc`

- Usual approach (as in `ols`) to correcting for over-optimism through bootstrap validation, now using 50 bootstrap resamples instead of 40.

```r
set.seed(432)
validate(mod_cc, B = 50)
```



```
          index.orig training    test optimism index.corrected  n
Dxy           0.3634   0.3655  0.3583   0.0072          0.3562 50
R2            0.1001   0.1007  0.0977   0.0029          0.0972 50
Intercept     0.0000   0.0000 -0.0196   0.0196         -0.0196 50
Slope         1.0000   1.0000  0.9873   0.0127          0.9873 50
Emax          0.0000   0.0000  0.0064   0.0064          0.0064 50
D             0.0592   0.0596  0.0578   0.0018          0.0574 50
U            -0.0005  -0.0005  0.0000  -0.0006          0.0000 50
Q             0.0598   0.0601  0.0577   0.0024          0.0574 50
B             0.1216   0.1215  0.1219  -0.0004          0.1220 50
g             0.6892   0.6933  0.6829   0.0105          0.6787 50
gp            0.0917   0.0918  0.0907   0.0011          0.0906 50
```

- Summaries we'll focus on here are `Dxy`, `R2` and `B`
- Remember that $C = 0.5 + \frac{Dxy}{2}$, so our validated C statistic would be $0.5 + (0.3562/2) = 0.6781$

ANOVA for `mod_cc`

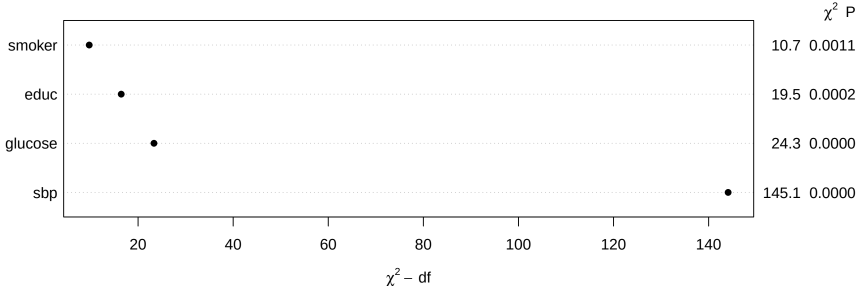
Model `mod_cc` uses 6 degrees of freedom.

```r
anova(mod_cc)
```

```
                Wald Statistics          Response: chd10

 Factor     Chi-Square d.f. P
 glucose     24.34     1    <.0001
 smoker      10.72     1    0.0011
 sbp        145.10     1    <.0001
 educ        19.45     3    0.0002
 TOTAL      208.87     6    <.0001
```

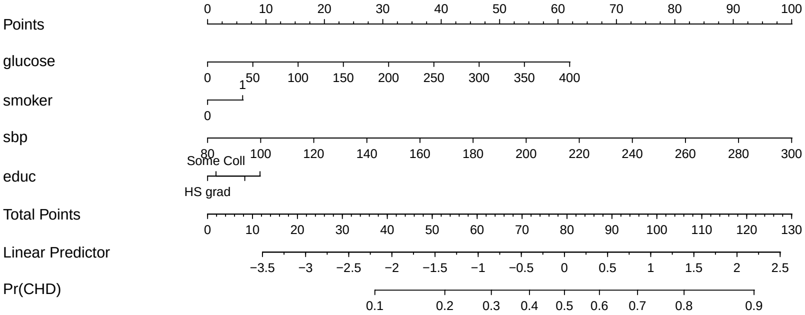
ANOVA for Model `mod_cc`

```
plot(anova(mod_cc))
```



Nomogram for `mod_cc`

```
plot(nomogram(mod_cc, fun = plogis,
            funlabel = "Pr(CHD)"))
```



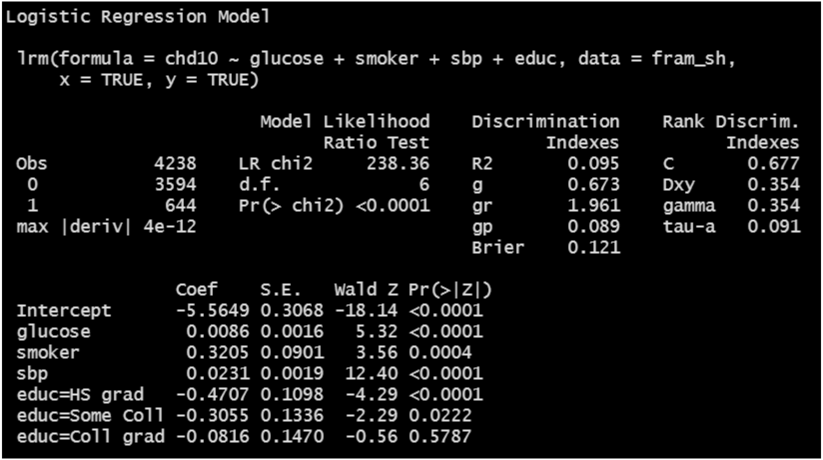
Section 3

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

Fit mod_si which is mod_cc after single imputation

```
d <- datadist(fram_sh)
options(datadist = "d")

mod_si <- lrm(chd10 ~ glucose + smoker + sbp + educ,
             data = fram_sh, x = TRUE, y = TRUE)
```

Model `mod_si` with single imputation



```
Logistic Regression Model

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

                       Model Likelihood     Discrimination    Rank Discrim.
                          Ratio Test           Indexes           Indexes
 Obs            4238    LR chi2     238.36    R2       0.095    C       0.677
  0             3594    d.f.             6    g        0.673    Dxy     0.354
  1              644    Pr(> chi2) <0.0001    gr       1.961    gamma   0.354
 max |deriv| 4e-12                            gp       0.089    tau-a   0.091
                                              Brier    0.121

                Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept      -5.5649 0.3068 -18.14 <0.0001
 glucose         0.0086 0.0016   5.32 <0.0001
 smoker          0.3205 0.0901   3.56 0.0004
 sbp             0.0231 0.0019  12.40 <0.0001
 educ=HS grad   -0.4707 0.1098  -4.29 <0.0001
 educ=Some Coll -0.3055 0.1336  -2.29 0.0222
 educ=Coll grad -0.0816 0.1470  -0.56 0.5787
```

Comparing the Coefficients (exponentiated)

- Comparing the slopes as odds ratios

```
round_half_up(exp(mod_cc$coefficients),3)
```

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

```
round_half_up(exp(mod_si$coefficients),3)
```

```
     Intercept          glucose           smoker              sbp 
         0.004            1.009            1.378            1.023 
educ=Some Coll educ=Coll grad 
         0.737            0.922 
```

Edited Summaries Comparing The Models

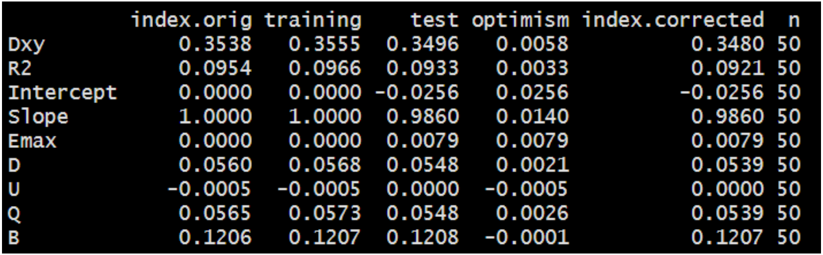
Summary	mod_si value	mod_cc value
Obs	4238	3753
0	3594	3174
1	644	579
Nagelkerke R^2	0.095	0.100
Brier Score	0.121	0.122
C	0.677	0.682
Dxy	0.354	0.363

- All of these results came from

```
mod_cc
mod_si
```

Validate mod_si Summary Statistics

```r
set.seed(432)
validate(mod_si, B = 50)
```

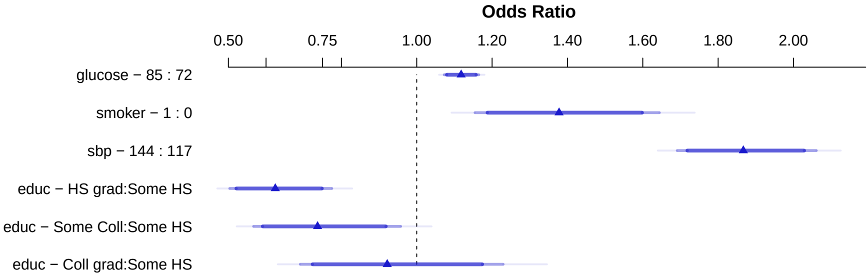


```
          index.orig training    test optimism index.corrected  n
Dxy           0.3538   0.3555  0.3496   0.0058          0.3480 50
R2            0.0954   0.0966  0.0933   0.0033          0.0921 50
Intercept     0.0000   0.0000 -0.0256   0.0256         -0.0256 50
Slope         1.0000   1.0000  0.9860   0.0140          0.9860 50
Emax          0.0000   0.0000  0.0079   0.0079          0.0079 50
D             0.0560   0.0568  0.0548   0.0021          0.0539 50
U            -0.0005  -0.0005  0.0000  -0.0005          0.0000 50
Q             0.0565   0.0573  0.0548   0.0026          0.0539 50
B             0.1206   0.1207  0.1208  -0.0001          0.1207 50
```

- Again, $C = 0.5 + \frac{Dxy}{2}$, so the corrected C statistic estimate will be 0.5 + (0.348/2) = 0.674

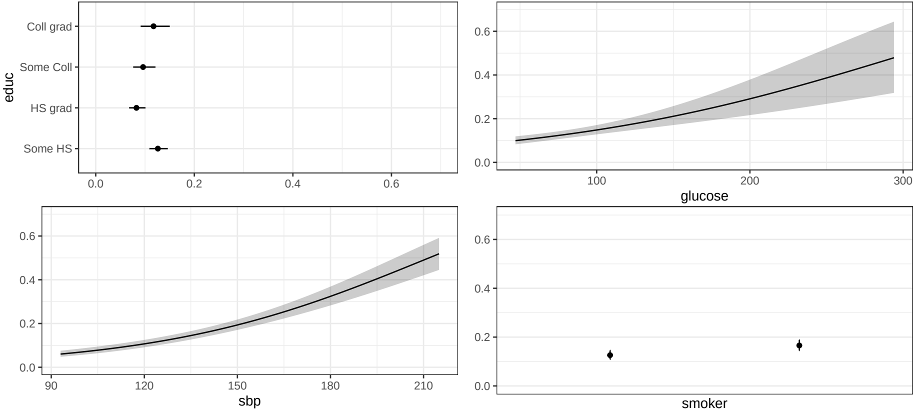
Plot of Effects using `mod_si`

```
plot(summary(mod_si))
```



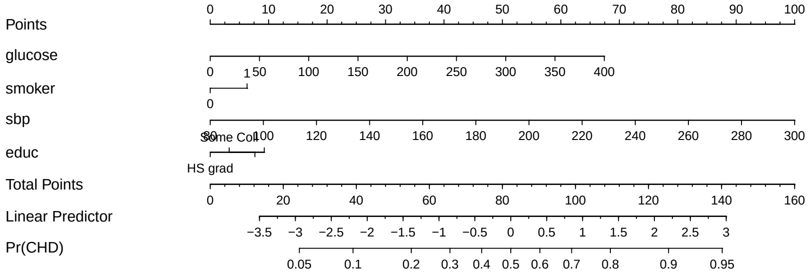
Predict results for `mod_si`

```r
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

Section 4

Using Multiple Imputation: The 4-predictor Model

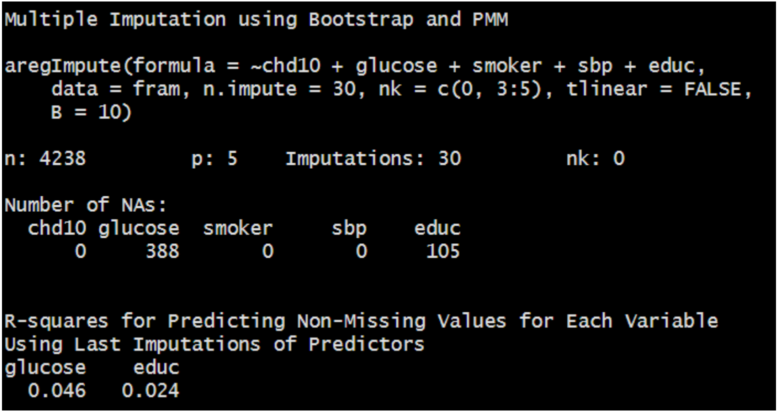
Fit the Imputation Model first

We'll use `aregImpute` here, and create 30 imputed sets.

```r
set.seed(432)
dd <- datadist(fram)
options(datadist = "dd")

fit_imp <-
    aregImpute(~ chd10 + glucose + smoker + sbp + educ,
               nk = c(0, 3:5), tlinear = FALSE, data = fram,
               B = 10, n.impute = 30)
```

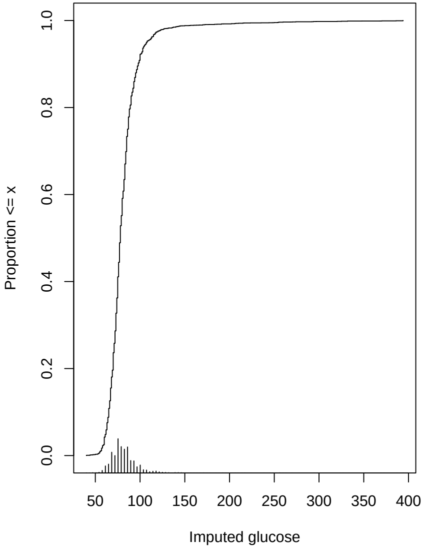
```
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
```

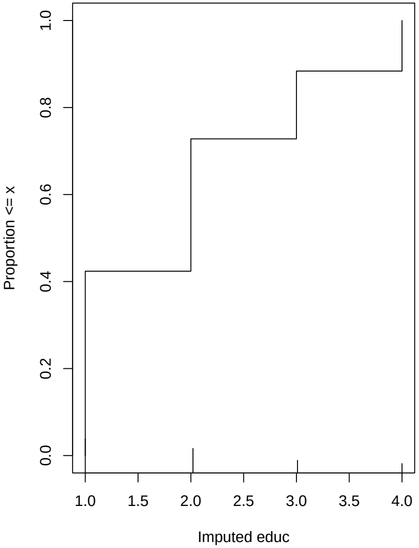
Imputation Results (abbreviated output)



```
Multiple Imputation using Bootstrap and PMM

aregImpute(formula = ~chd10 + glucose + smoker + sbp + educ,
    data = fram, n.impute = 30, nk = c(0, 3:5), tlinear = FALSE,
    B = 10)

n: 4238         p: 5    Imputations: 30         nk: 0

Number of NAs:
  chd10 glucose  smoker     sbp    educ
      0     388       0       0     105


R-squares for Predicting Non-Missing Values for Each Variable
Using Last Imputations of Predictors
glucose    educ
  0.046   0.024
```

Multiply Imputed Values, via `plot(fit_imp)`





What do we need to do our multiple imputation?

- Imputation Model

```
fit_imp <-
    aregImpute(~ chd10 + glucose + smoker + sbp + educ,
               nk = c(0, 3:5), tlinear = FALSE, data = fram,
               B = 10, n.impute = 30)
```

- Outcome Model will be of the following form...

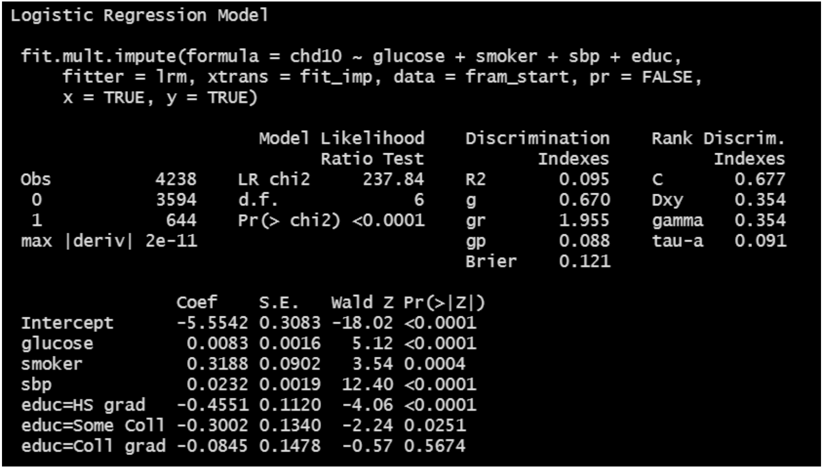
```
lrm(chd10 ~ glucose + smoker + sbp + educ,
    x = TRUE, y = TRUE)
```

Fitting mod_mi (mod_cc with multiple imputation)

```
mod_mi <-
    fit.mult.impute(chd10 ~ glucose + smoker + sbp + educ,
                    fitter = lrm, xtrans = fit_imp,
                    data = fram_start, 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)
- `pr = FALSE` avoids a long printout we don't need

Model mod_mi with multiple imputation



```
Logistic Regression Model

 fit.mult.impute(formula = chd10 ~ glucose + smoker + sbp + educ,
     fitter = lrm, xtrans = fit_imp, data = fram_start, pr = FALSE,
     x = TRUE, y = TRUE)

                        Model Likelihood    Discrimination     Rank Discrim.
                              Ratio Test           Indexes           Indexes
 Obs           4238    LR chi2     237.84    R2       0.095     C       0.677
  0            3594    d.f.             6    g        0.670     Dxy     0.354
  1             644    Pr(> chi2) <0.0001    gr       1.955     gamma   0.354
 max |deriv| 2e-11                           gp       0.088     tau-a   0.091
                                             Brier    0.121

                Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept      -5.5542 0.3083 -18.02 <0.0001
 glucose         0.0083 0.0016   5.12 <0.0001
 smoker          0.3188 0.0902   3.54 0.0004
 sbp             0.0232 0.0019  12.40 <0.0001
 educ=HS grad   -0.4551 0.1120  -4.06 <0.0001
 educ=Some Coll -0.3002 0.1340  -2.24 0.0251
 educ=Coll grad -0.0845 0.1478  -0.57 0.5674
```

Comparing the Coefficients (exponentiated)

- I'll just compare the two models using imputation...

```
round_half_up(exp(mod_mi$coefficients),3)
```

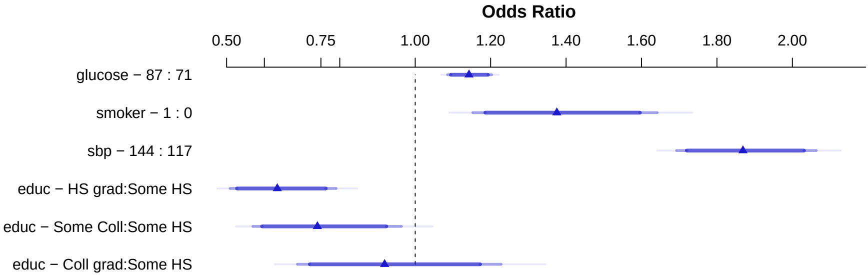
```
     Intercept        glucose         smoker            sbp 
         0.004          1.008          1.376          1.023 
educ=Some Coll educ=Coll grad 
         0.741          0.919 
```

```
round_half_up(exp(mod_si$coefficients),3)
```

```
     Intercept        glucose         smoker            sbp 
         0.004          1.009          1.378          1.023 
educ=Some Coll educ=Coll grad 
         0.737          0.922 
```

Plot of Effects using `mod_mi`

```
plot(summary(mod_mi))
```



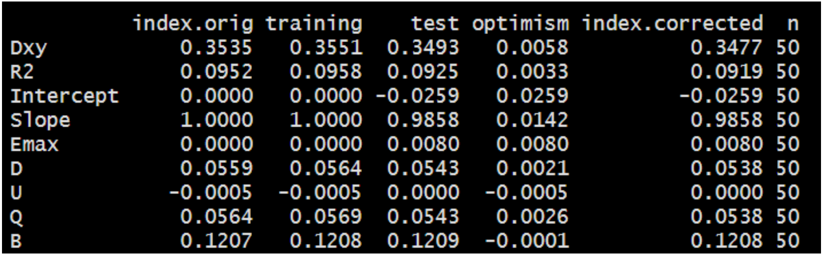
Edited Summaries Comparing Our 3 Models

Summary	`mod_mi` value	`mod_si` value	`mod_cc` value
Obs	4238	4238	3753
0	3594	3594	3174
1	644	644	579
Nagelkerke R^2	0.095	0.095	0.100
Brier Score	0.121	0.121	0.122
C	0.677	0.677	0.682
Dxy	0.354	0.354	0.363

- It's just a coincidence that the `mod_mi` and `mod_si` values are identical to the level of precision provided in this table.
- What might cause the values to look meaningfully different?

Validate `mod_mi` Summary Statistics

```r
set.seed(432)
validate(mod_mi, B = 50)
```

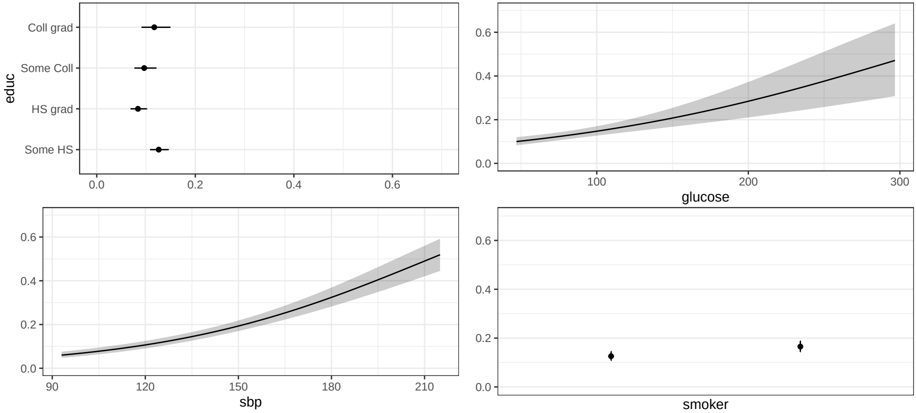


```
          index.orig training    test optimism index.corrected  n
Dxy           0.3535   0.3551  0.3493   0.0058          0.3477 50
R2            0.0952   0.0958  0.0925   0.0033          0.0919 50
Intercept     0.0000   0.0000 -0.0259   0.0259         -0.0259 50
Slope         1.0000   1.0000  0.9858   0.0142          0.9858 50
Emax          0.0000   0.0000  0.0080   0.0080          0.0080 50
D             0.0559   0.0564  0.0543   0.0021          0.0538 50
U            -0.0005  -0.0005  0.0000  -0.0005          0.0000 50
Q             0.0564   0.0569  0.0543   0.0026          0.0538 50
B             0.1207   0.1208  0.1209  -0.0001          0.1208 50
```

- Optimism-corrected C statistic estimate is $0.5 + (0.3477/2) = 0.674$

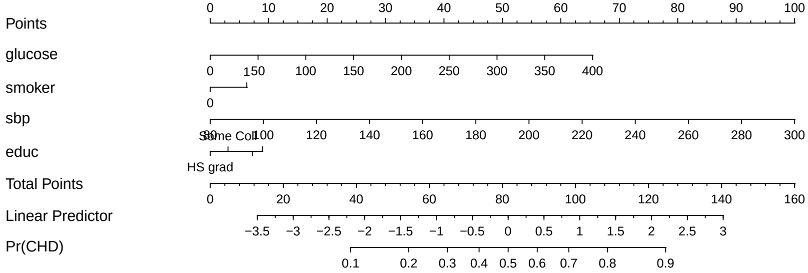
Predict results for `mod_mi`

```
ggplot(Predict(mod_mi, fun = plogis))
```



Nomogram for `mod_mi`

```
plot(nomogram(mod_mi, fun = plogis,
            funlabel = "Pr(CHD)"))
```

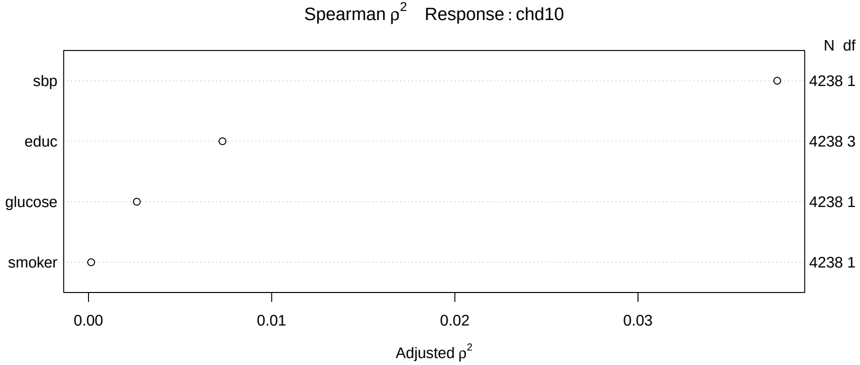


Section 5

Considering Non-Linear Terms

Spearman ρ^2 Plot

```
plot(spearman2(chd10 ~ glucose + smoker + sbp + educ,
               data = fram_sh))
```



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…

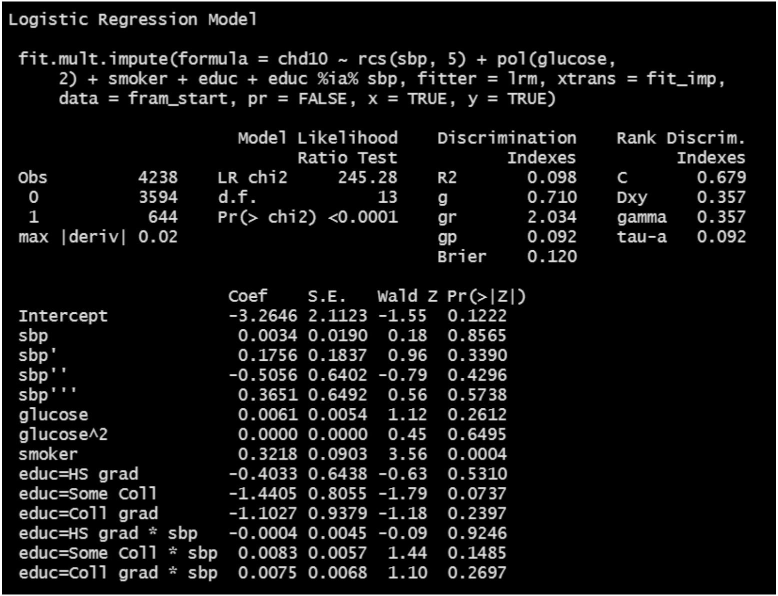
- I'll just show the results including the multiple imputation, since if you can get those, you should have little difficulty instead applying the single imputation or the complete case analysis.

mod_big incorporating multiple imputation

Our mod_big will incorporate several non-linear terms.

```r
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, x = TRUE, y = TRUE,
      pr = FALSE)
```

The mod_big model with non-linear terms

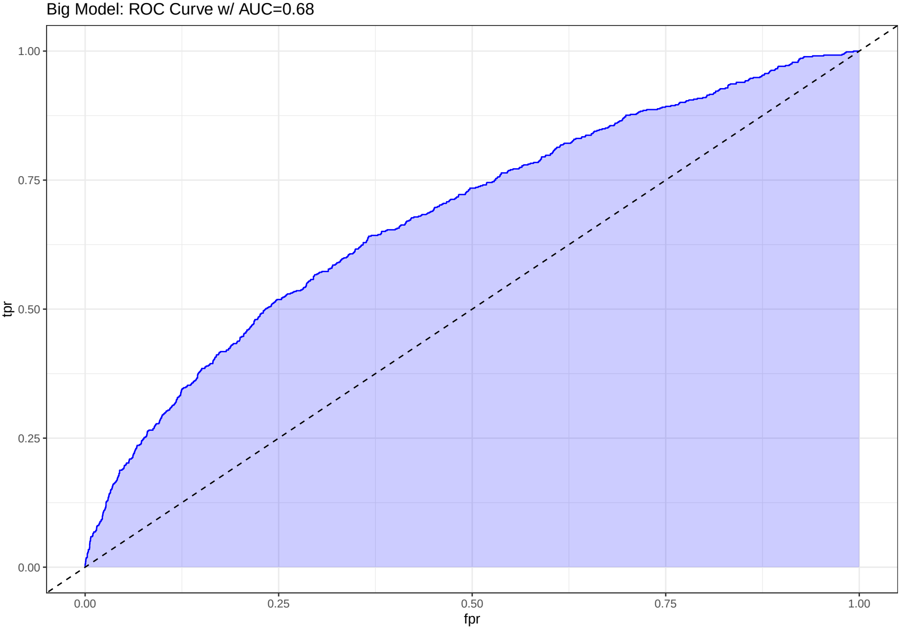


```
Logistic Regression Model

 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, x = TRUE, y = TRUE)

                       Model Likelihood    Discrimination    Rank Discrim.
                          Ratio Test           Indexes          Indexes
 Obs          4238    LR chi2     245.28    R2       0.098    C       0.679
  0           3594    d.f.            13    g        0.710    Dxy     0.357
  1            644    Pr(> chi2) <0.0001    gr       2.034    gamma   0.357
 max |deriv| 0.02                           gp       0.092    tau-a   0.092
                                            Brier    0.120

                       Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept             -3.2646 2.1123 -1.55  0.1222
 sbp                    0.0034 0.0190  0.18  0.8565
 sbp'                   0.1756 0.1837  0.96  0.3390
 sbp''                 -0.5056 0.6402 -0.79  0.4296
 sbp'''                 0.3651 0.6492  0.56  0.5738
 glucose                0.0061 0.0054  1.12  0.2612
 glucose^2              0.0000 0.0000  0.45  0.6495
 smoker                 0.3218 0.0903  3.56  0.0004
 educ=HS grad          -0.4033 0.6438 -0.63  0.5310
 educ=Some Coll        -1.4405 0.8055 -1.79  0.0737
 educ=Coll grad        -1.1027 0.9379 -1.18  0.2397
 educ=HS grad * sbp    -0.0004 0.0045 -0.09  0.9246
 educ=Some Coll * sbp   0.0083 0.0057  1.44  0.1485
 educ=Coll grad * sbp   0.0075 0.0068  1.10  0.2697
```

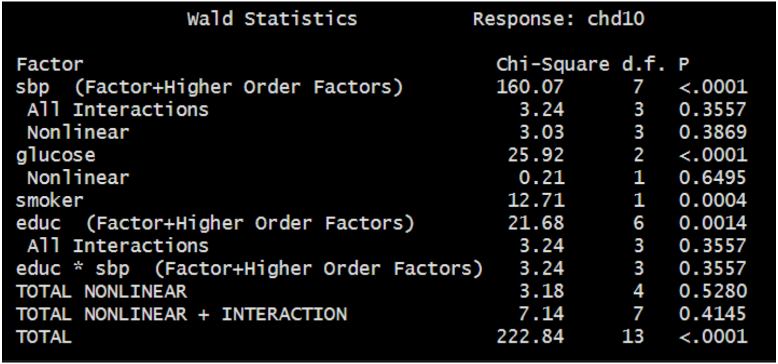
mod_big vs. mod_mi comparison

Summary	mod_big	mod_mi
Obs	4238	4238
0	3594	3594
1	644	644
Nagelkerke R^2	0.098	0.095
Brier Score	0.120	0.121
C	0.679	0.677
Dxy	0.357	0.354

ROC Curve for `mod_big`



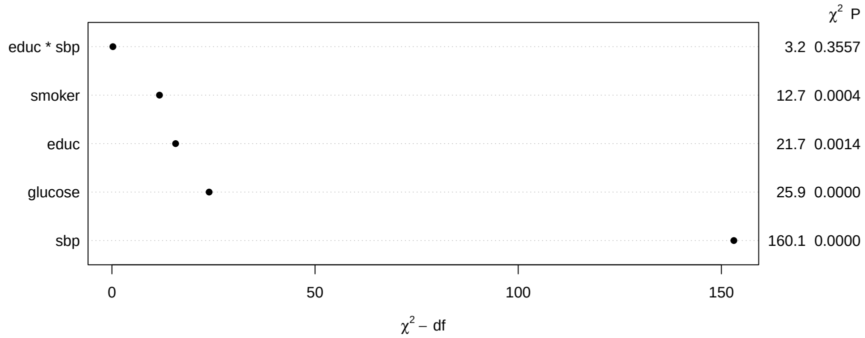
What does ANOVA suggest about the fit?



```
                Wald Statistics          Response: chd10

 Factor                                          Chi-Square d.f. P
 sbp  (Factor+Higher Order Factors)              160.07      7   <.0001
  All Interactions                                 3.24      3   0.3557
  Nonlinear                                        3.03      3   0.3869
 glucose                                          25.92      2   <.0001
  Nonlinear                                        0.21      1   0.6495
 smoker                                           12.71      1   0.0004
 educ  (Factor+Higher Order Factors)              21.68      6   0.0014
  All Interactions                                 3.24      3   0.3557
 educ * sbp  (Factor+Higher Order Factors)         3.24      3   0.3557
 TOTAL NONLINEAR                                   3.18      4   0.5280
 TOTAL NONLINEAR + INTERACTION                     7.14      7   0.4145
 TOTAL                                           222.84     13   <.0001
```

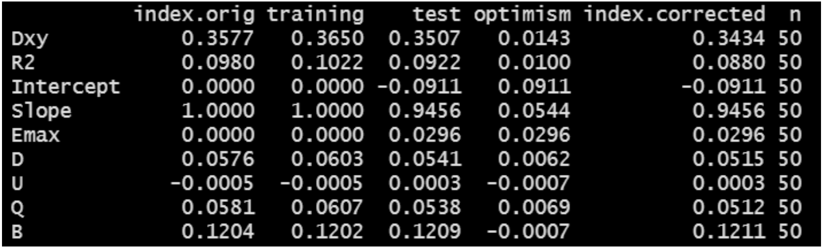
plot(anova(mod_big)) (model includes 13 df)

```
plot(anova(mod_big))
```



Validate mod_big Summary Statistics

```r
set.seed(432)
validate(mod_big, B = 50)
```

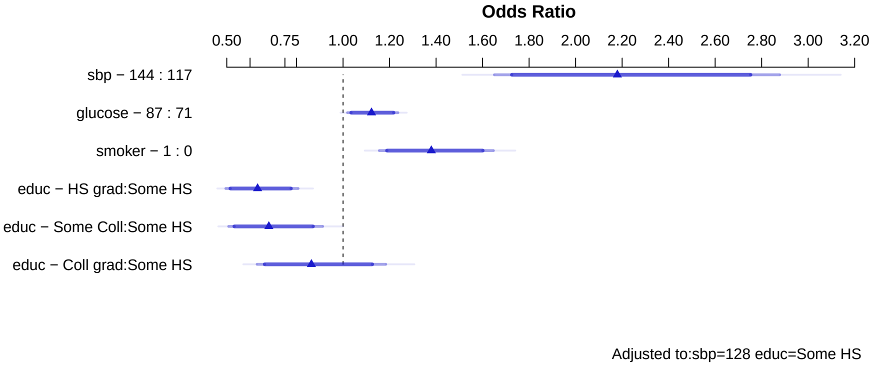


```
          index.orig training    test optimism index.corrected  n
Dxy           0.3577   0.3650  0.3507   0.0143          0.3434 50
R2            0.0980   0.1022  0.0922   0.0100          0.0880 50
Intercept     0.0000   0.0000 -0.0911   0.0911         -0.0911 50
Slope         1.0000   1.0000  0.9456   0.0544          0.9456 50
Emax          0.0000   0.0000  0.0296   0.0296          0.0296 50
D             0.0576   0.0603  0.0541   0.0062          0.0515 50
U            -0.0005  -0.0005  0.0003  -0.0007          0.0003 50
Q             0.0581   0.0607  0.0538   0.0069          0.0512 50
B             0.1204   0.1202  0.1209  -0.0007          0.1211 50
```

- Optimism-Corrected $C = 0.5 + (.3434/2) = .672$

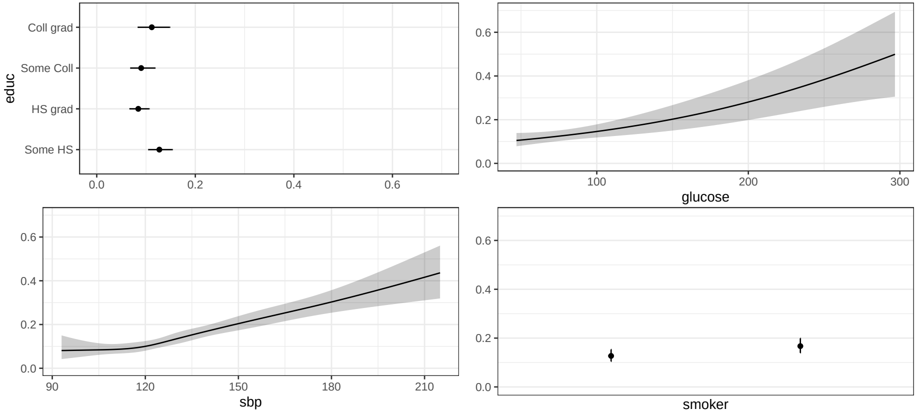
Plot of Effects using `mod_big`

```
plot(summary(mod_big))
```



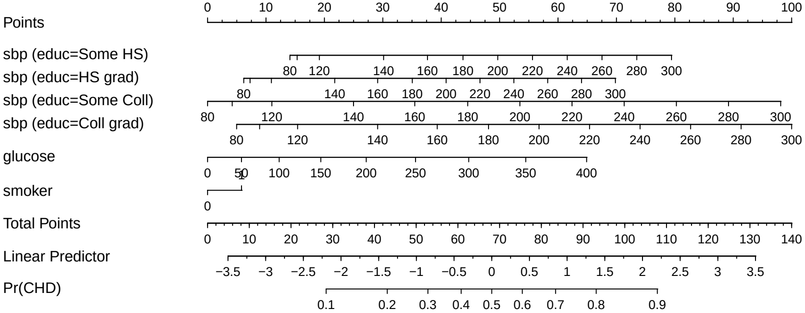
Predict results for `mod_big`

```
ggplot(Predict(mod_big, fun = plogis))
```



Nomogram for `mod_big`

```
plot(nomogram(mod_big, fun = plogis, funlabel = "Pr(CHD)"))
```



Next Time

- Variable Selection in Linear Regression
- Ridge Regression and the Lasso
- K-fold Cross-Validation in a Linear Regression Model