432 Class 11 Slides

github.com/THOMASELOVE/2020-432

2020-02-20

Setup

```
library(magrittr); library(janitor); library(here)

library(ROCR) # to draw ROC curves
library(naniar)
library(simputation)
library(broom)
library(rms) # note: also loads Hmisc
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

```
fram_raw <- read_csv(here("data/framingham.csv")) %>%
    clean_names()
```

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

This particular data set has been used by lots of people, in many different settings, and variations of it are all over the internet. 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, male, age, educ, everything()) %>%
    select(-education)
```

Data Descriptions (first 10 variables)

The variables describe n = 4238 adult subjects who were examined at baseline and then followed for ten years to see if they developed incident coronary heart disease during that time.

Variable	Description
`subj_id`	identifying code added by Dr. Love
`male`	1 = subject is male, else 0
`age`	in years (range is 32 to 70)
`educ`	four-level factor: educational attainment
`smoker`	1 = current smoker at time of examination, else 0
`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

Data Descriptions (Other 7 variables)

Variable	Description
chol	total cholesterol (mg/dl)
sbp	systolic blood pressure (mm Hg)
dbp	diastolic blood pressure (mm Hg)
bmi	body mass index in kg/m^2
hrate	heart rate in beats per minute
glucose	blood glucose level in mg/dl
chd10	1 = coronary heart disease in next 10 years

Our outcome is `chd10`, which has no missing data here...

```
fram %>% tabyl(chd10) %>% adorn_pct_formatting(digits = 2)
```

```
 chd10    n percent
     0 3594  84.80%
     1  644  15.20%
```

Any Missing Data?

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

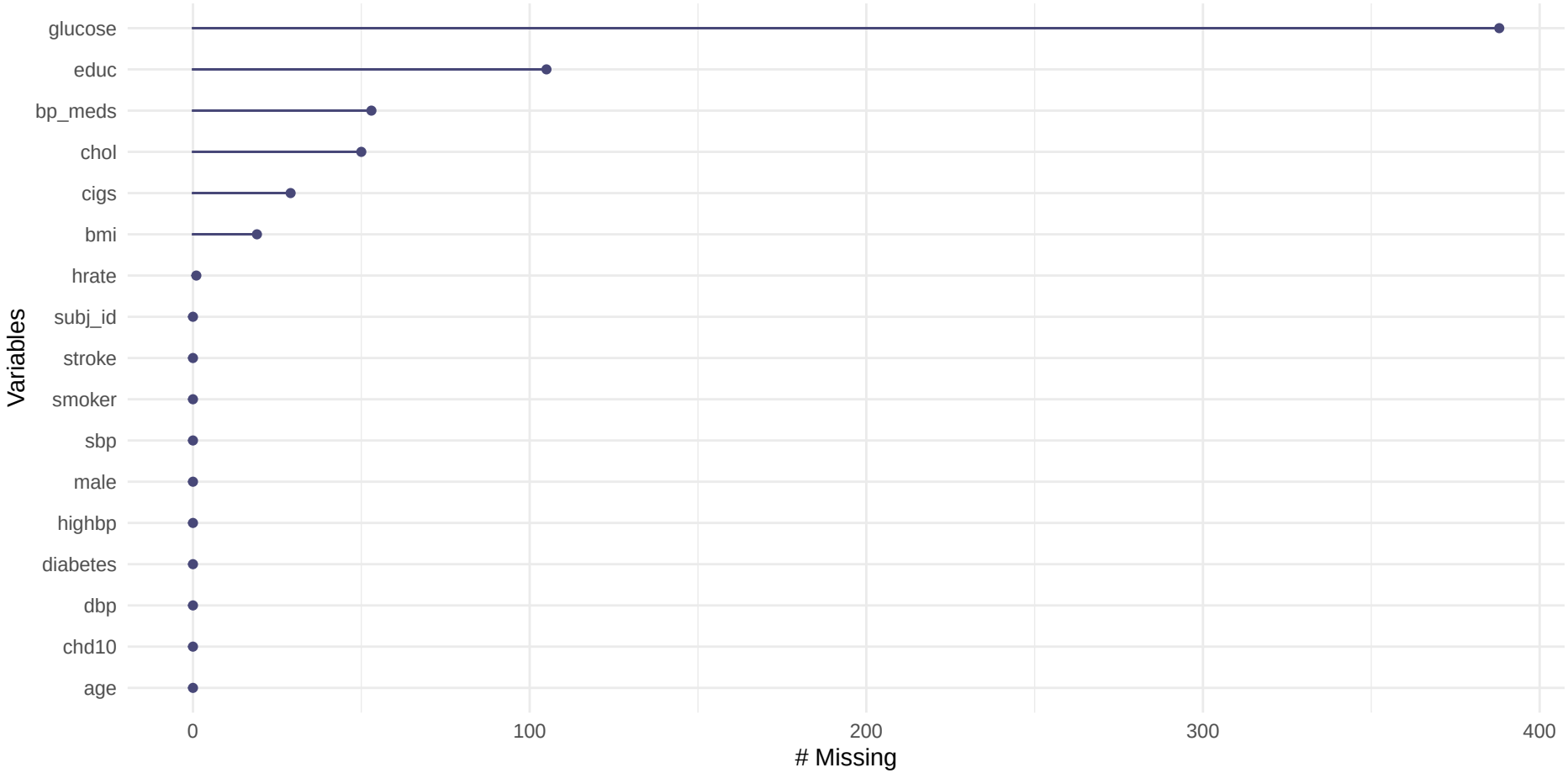
```
[1] 3656
```

```
[1] 86.26711
```

- 3656 (86.3%) of the 4238 subjects in the `fram` data are complete.
- The remaining 582 observations have something missing.

Which variables are missing data?

```
gg_miss_var(fram)
```



Counts of Missing Data, by Variable

```
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
```

fram_cc = "Complete Cases" Data Set

```
fram_cc <- fram %>% drop_na()

dim(fram)
```

```
[1] 4238   17
```

```
n_case_miss(fram)
```

```
[1] 582
```

```
dim(fram_cc)
```

```
[1] 3656   17
```

```
n_case_miss(fram_cc)
```

```
[1] 0
```

Simple Imputation

We need to impute:

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

```
fram_sh <- bind_shadow(fram)

set.seed(2020432)
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) %>%
    tbl_df()
```

Sanity Check

```
n_case_miss(fram_sh)
```

```
[1] 0
```

Check multi-categorical simple 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?

Goals today

1. Use `lrm` to fit a four-predictor logistic regression model to predict `chd10` using `glucose`, `smoker`, `sbp` and `educ`
   a. on the complete cases (`fram_cc`)
   b. accounting for missingness via simple imputation (`fram_sh`)
   c. accounting for missingness via multiple imputation
2. Consider adding some non-linear terms to the "four-predictor" models, and refit.

Fitting a Four-Predictor Model (Complete Cases)

A "Four Predictor" model

First, we'll use the `fram` data set (which includes missing values) and perform a complete-case analysis.

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

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

modA (output slide 1)

```
Frequencies of Missing Values Due to Each Variable
  chd10 glucose  smoker     sbp    educ
      0     388       0       0     105

Logistic Regression Model

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

modA (output slide 2: coefficients)

	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

modA (output slide 3: summaries)

Obs	3753
0	3174
1	579
max |deriv|	2e-09

Model Likelihood Ratio Test

LR chi2	223.29
d.f.	6
Pr(> chi2)	<0.0001

Discrimination Indexes

R2	0.100
g	0.689
gr	1.992
gp	0.092
Brier	0.122

Rank Discrimination Indexes

C	0.682
Dxy	0.363
gamma	0.364
tau-a	0.095

We'll walk through the meaning of this output in the next few slides. Also, see section 10.14 in the Course Notes.

Deconstructing the `modA` summaries

```
                          Model Likelihood Ratio Test
Obs            3753           LR chi2     223.29
 0             3174           d.f.             6
 1              579           Pr(> chi2) <0.0001
max |deriv| 2e-09
```

- `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.
- A likelihood ratio test (drop in deviance test) subtracting the residual deviance from the null deviance to obtain the Likelihood Ratio χ^2 statistic, subtracting residual df from null df to obtain degrees of freedom, and comparing the resulting test statistic to a χ^2 distribution with the appropriate degrees of freedom to determine a p value.

Deconstructing the `modA` summaries

```
Discrimination Indexes          Rank Discrimination 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.

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 10 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.

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 `modA` run with `lrm`, we can use:

```
1 - (modA$deviance[2] / modA$deviance[1])
```

```
[1] 0.069174
```

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

I encourage you to spend your time thinking about misclassification rates, more than R^2 in logistic regression.

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.

There's a nice decomposition of the Brier score into calibration and discrimination.

Receiver Operating Characteristic Curve Analysis

One way to assess the predictive accuracy within the model development sample in a logistic regression is to consider an 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 10.9 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:

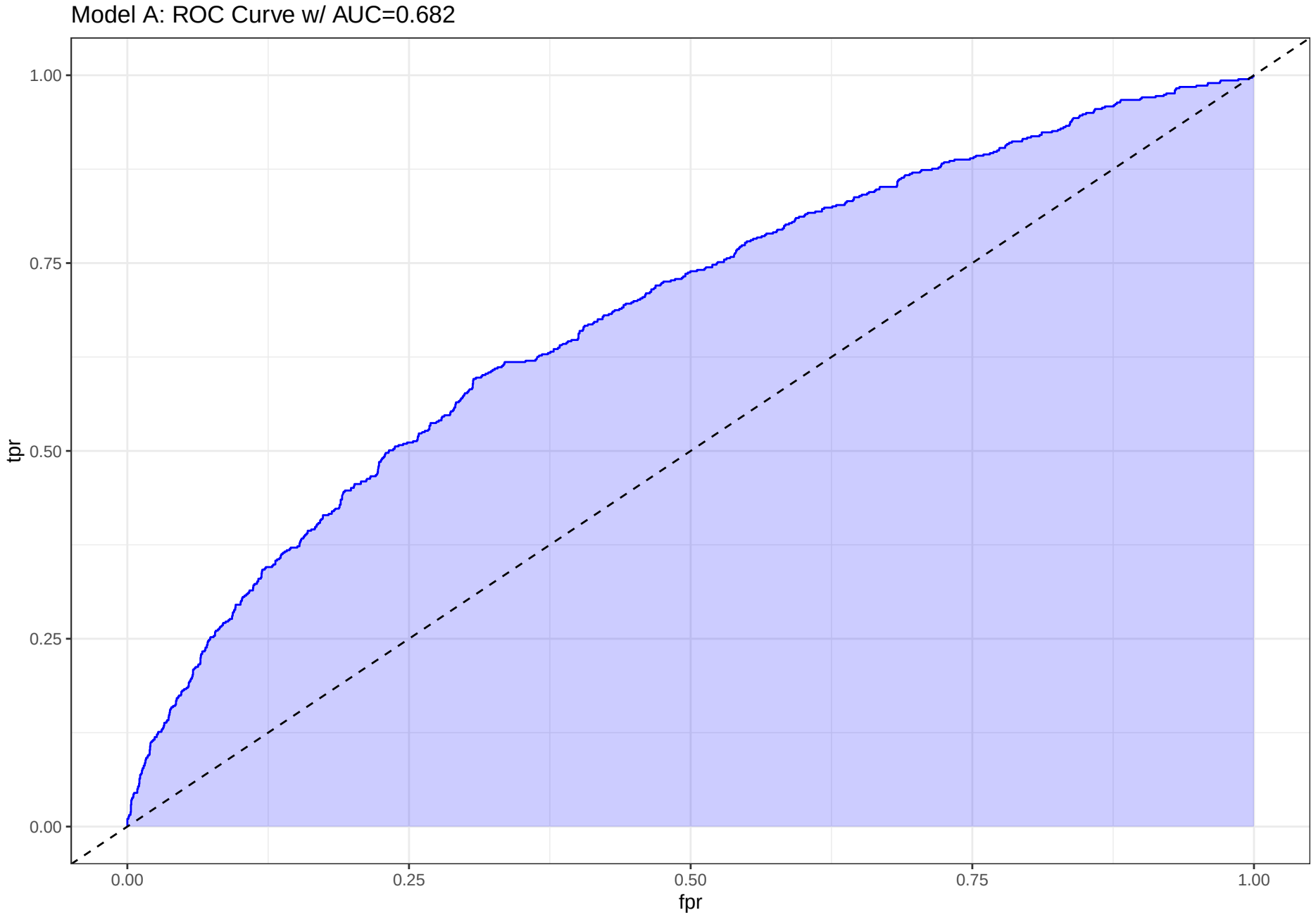
$$C = 0.5 + \frac{d}{2}, d = 2(C - .5)$$

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.

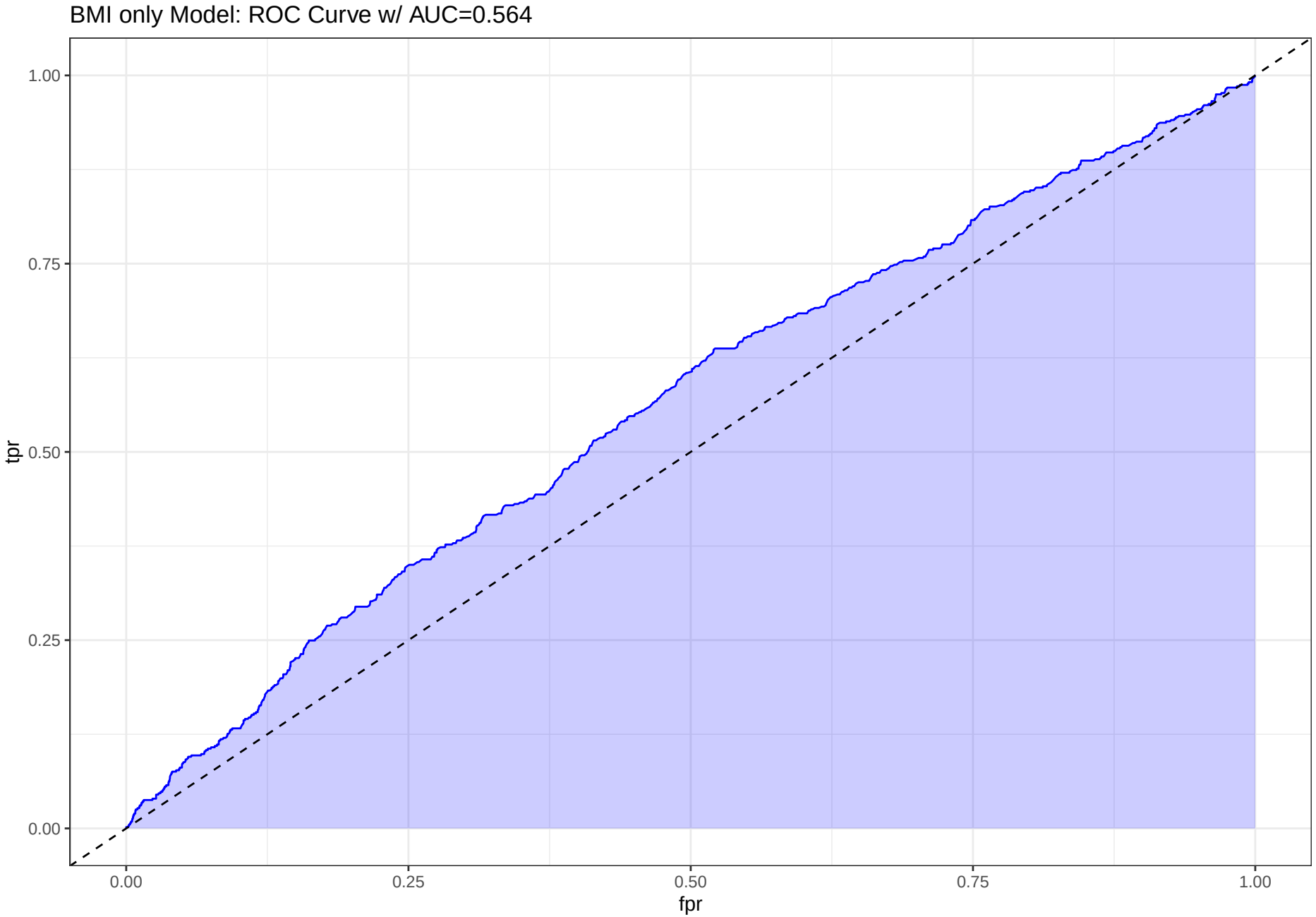
ROC Curve for our modA



Code for Previous Slide

```
## requires ROCR package
prob <- predict(modA, type="fitted")
pred <- prediction(prob, fram$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 (`bmi` only)



Validate Summary Statistics for modA

```
set.seed(2020)
validate(modA, B = 50)
```

Edited Results to fit on the screen. . .

	index.orig	training	test	optimism	index.corrected	n
Dxy	0.3634	0.3675	0.3584	0.0091	0.3543	50
R2	0.1001	0.1019	0.0979	0.0040	0.0961	50
B	0.1216	0.1217	0.1219	-0.0001	0.1218	50

Remember that $C = 0.5 + \frac{Dxy}{2}$.

ANOVA for modA

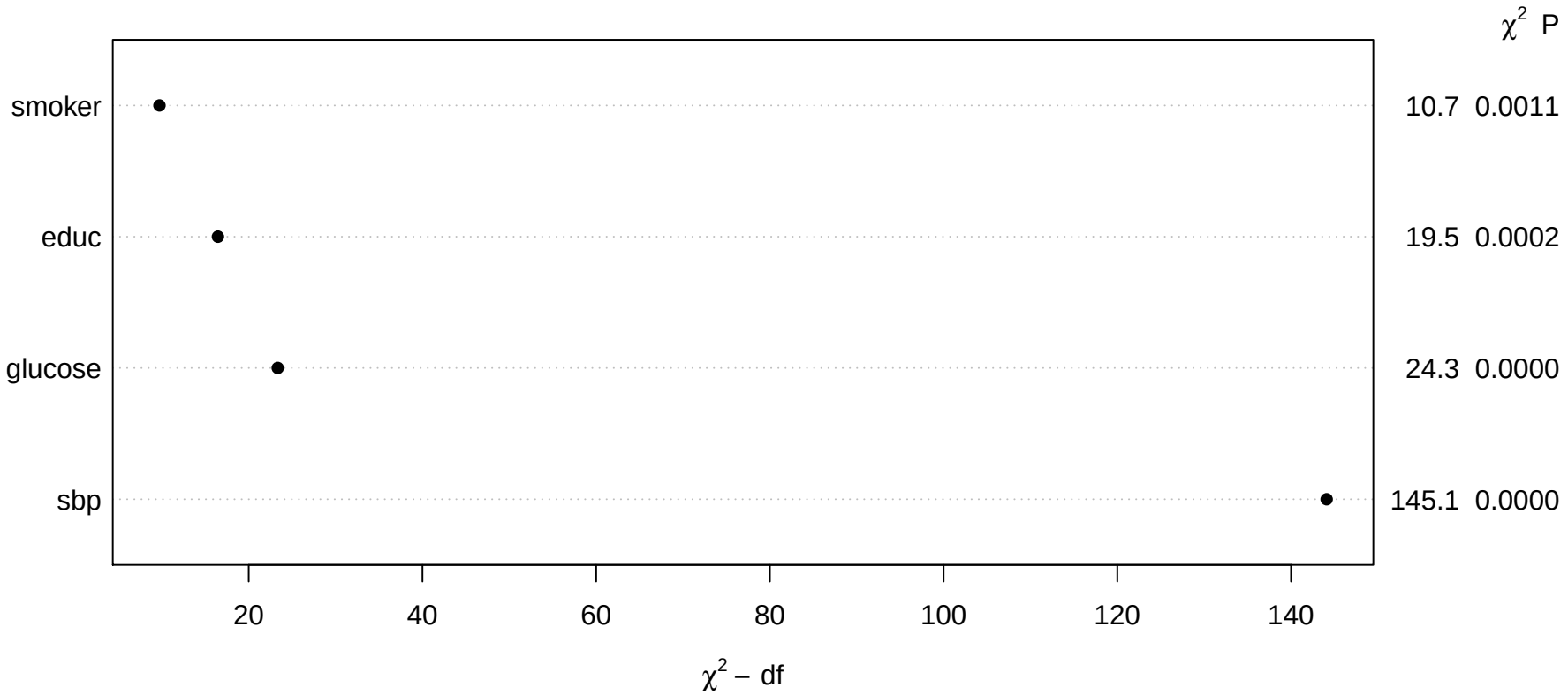
Model modA uses 6 degrees of freedom.

```
anova(modA)
```

```
                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
```

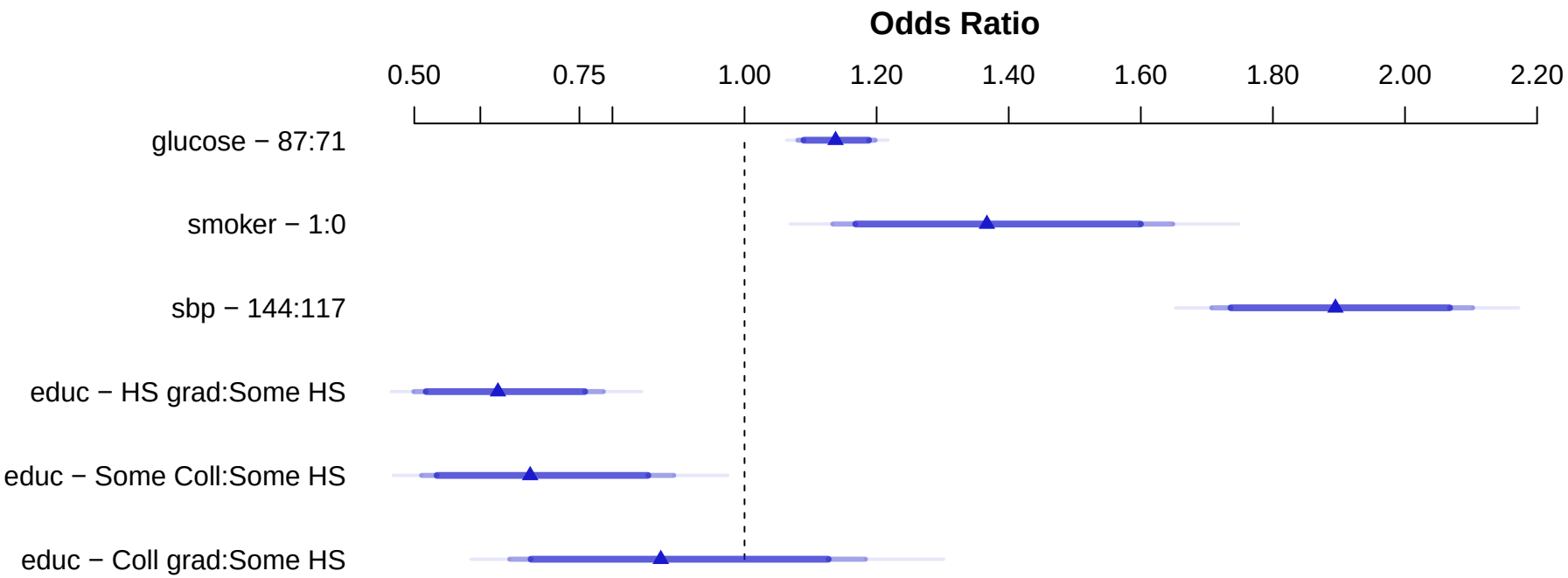
ANOVA for Model `modA`

```
plot(anova(modA))
```



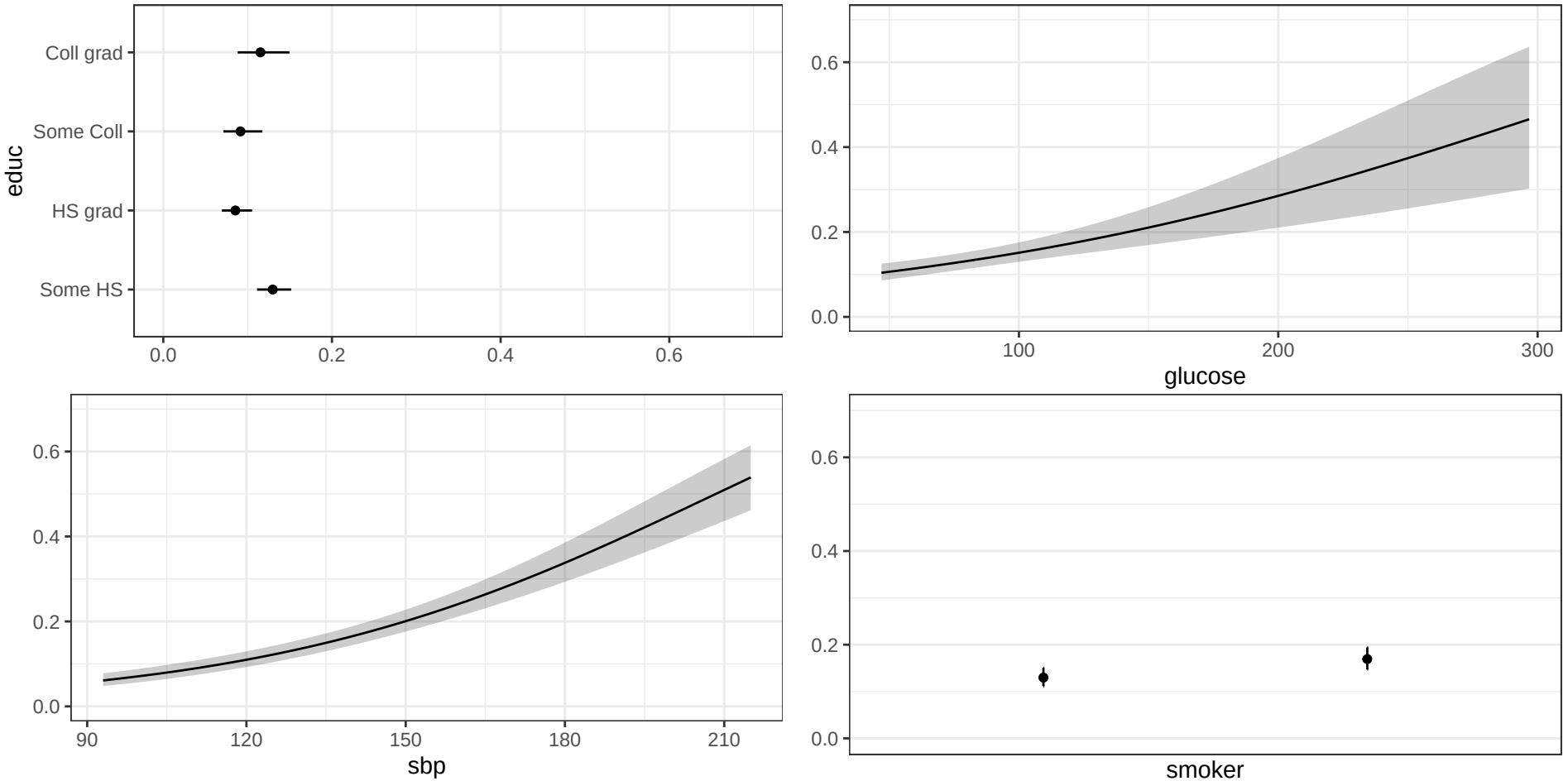
Plot of Effects using `modA`

```
plot(summary(modA))
```



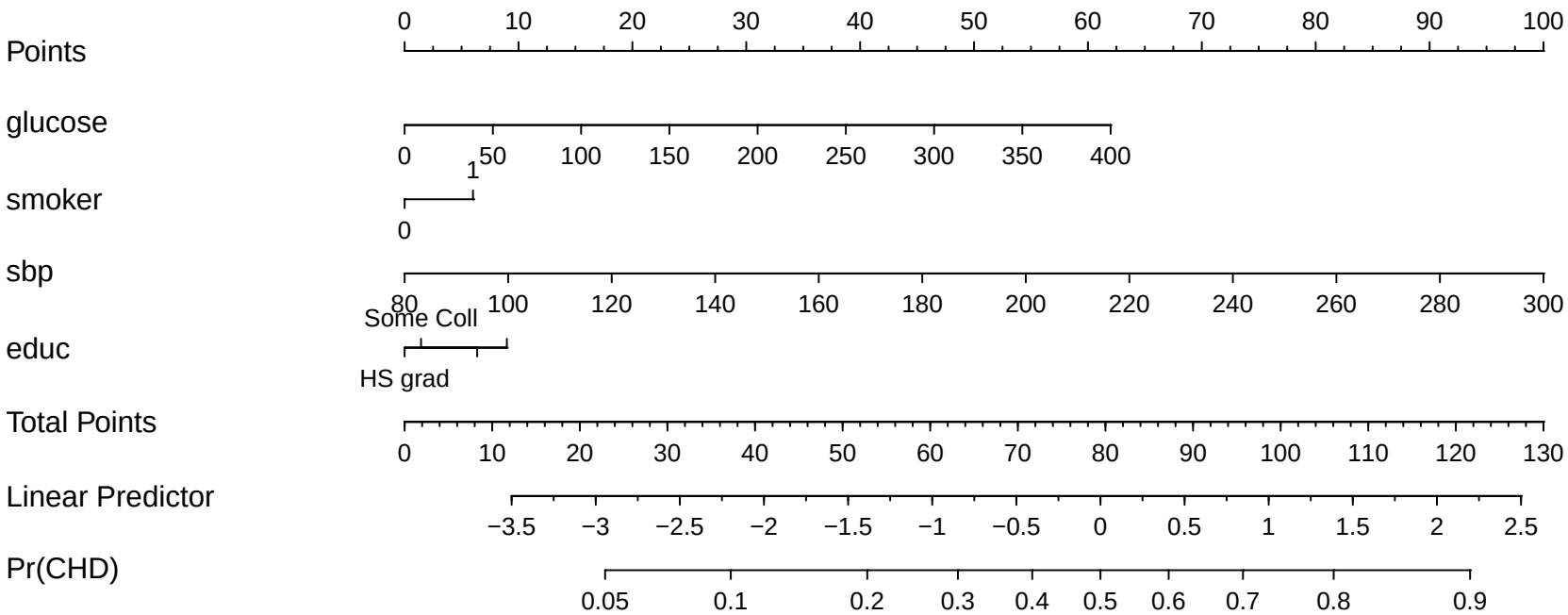
Predict results for `modA`

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



Nomogram for modA

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



Fitting a Four-Predictor Model (Single Imputation)

Fit modB which is modA after single imputation

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

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

Coefficients for Models B and A

- These indicate changes in the log odds ratio for `chd10`.
- Still would need to exponentiate to get odds ratios.

Term	B: Coeff.	B: SE	A: Coeff.	A: SE
Intercept	-5.565	0.307	-5.562	0.322
glucose	0.009	0.002	0.008	0.002
smoker	0.321	0.090	0.313	0.096
sbp	0.023	0.002	0.024	0.002
educ=HS grad	-0.471	0.110	-0.467	0.116
educ=Some Coll	-0.306	0.134	-0.392	0.142
educ=Coll grad	-0.082	0.147	-0.136	0.155

Edited Summaries Comparing Model B to Model A

Summary	modB value	modA 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

```
modA
modB
```

Validate modB Summary Statistics

```
set.seed(2020)
validate(modB, B = 50)
```

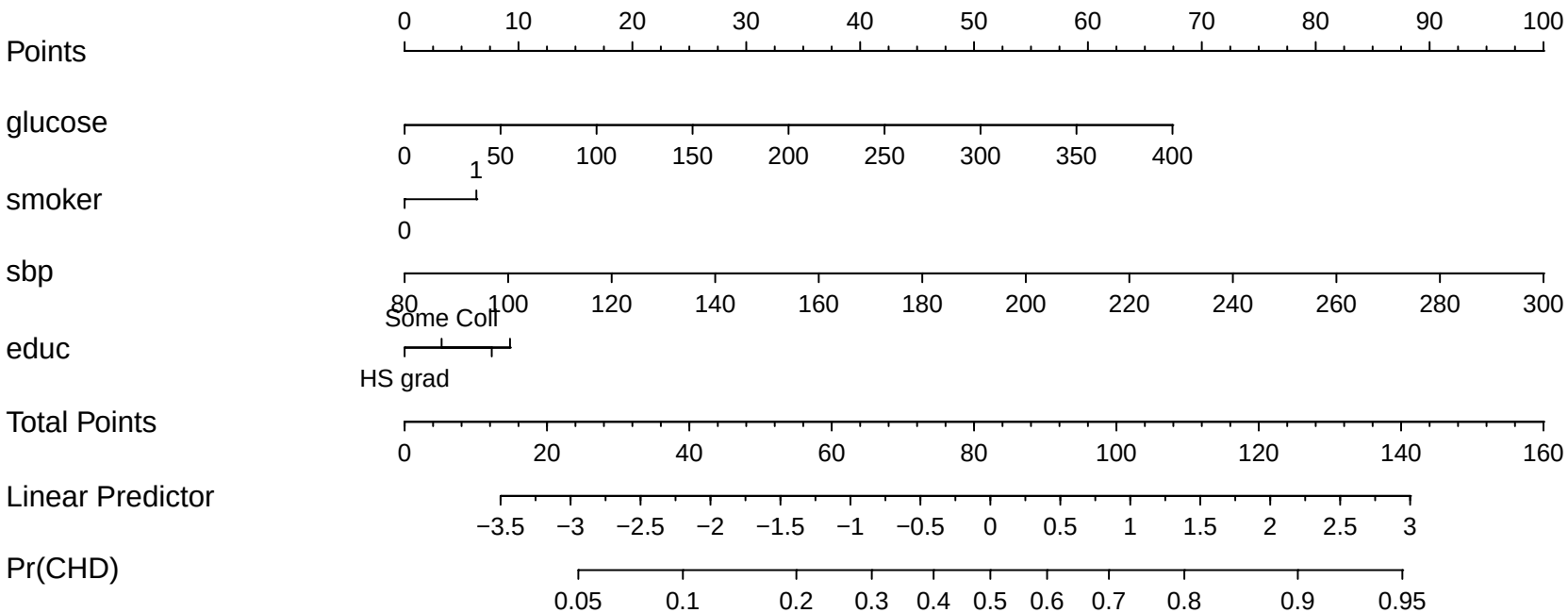
Edited Results to fit on the screen...

	index.orig	training	test	optimism	index.corrected	n
Dxy	0.3538	0.3526	0.3494	0.0033	0.3506	50
R2	0.0954	0.0951	0.0932	0.0019	0.0934	50
B	0.1206	0.1206	0.1208	-0.0002	0.1208	50

Again, remember that $C = 0.5 + \frac{Dxy}{2}$.

Nomogram for modB

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



Fitting a Four-Predictor Model (Multiple Imputation)

Fit Multiple Imputation Model

We'll use `aregImpute` here.

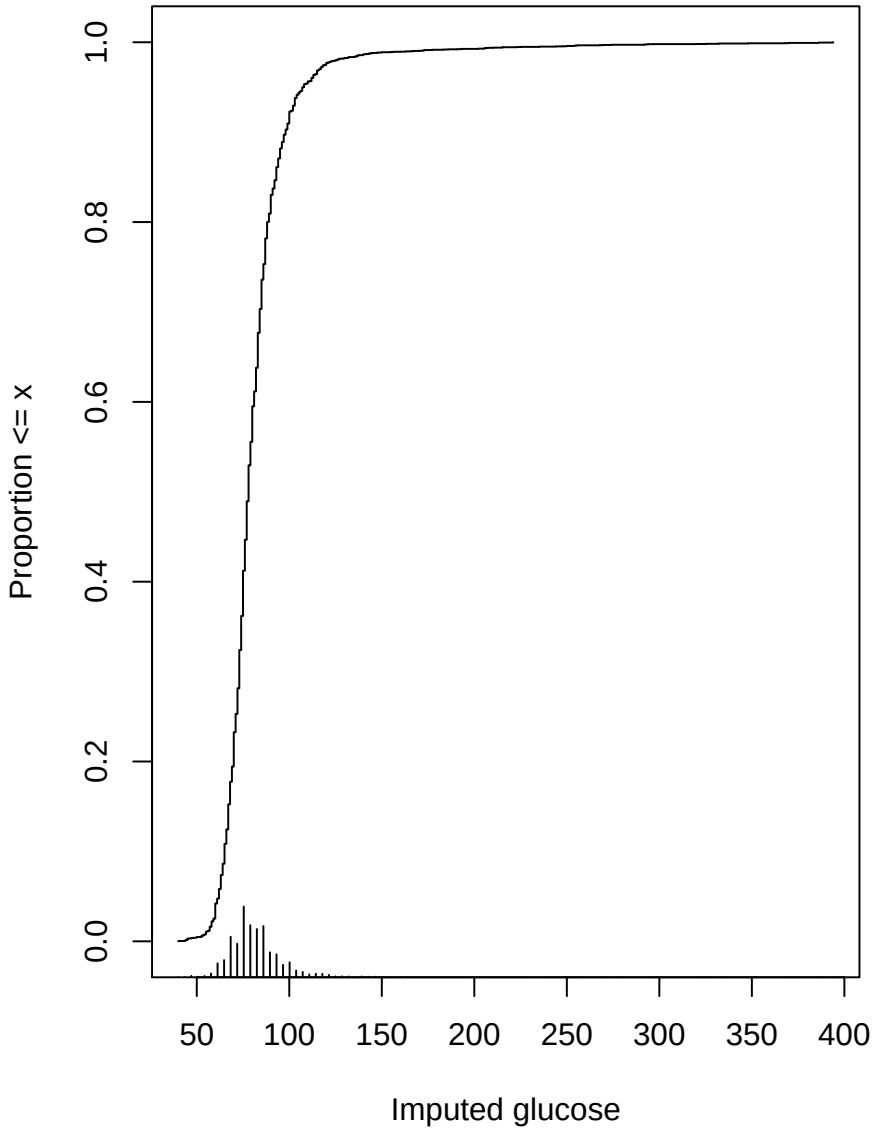
```
set.seed(4322020)
dd <- datadist(fram)
options(datadist = "dd")

fit_imp <-
    aregImpute(~ chd10 + glucose + smoker + sbp + educ,
               nk = c(0, 3:5), tlinear = FALSE, data = fram,
               B = 10, n.impute = 50)
```

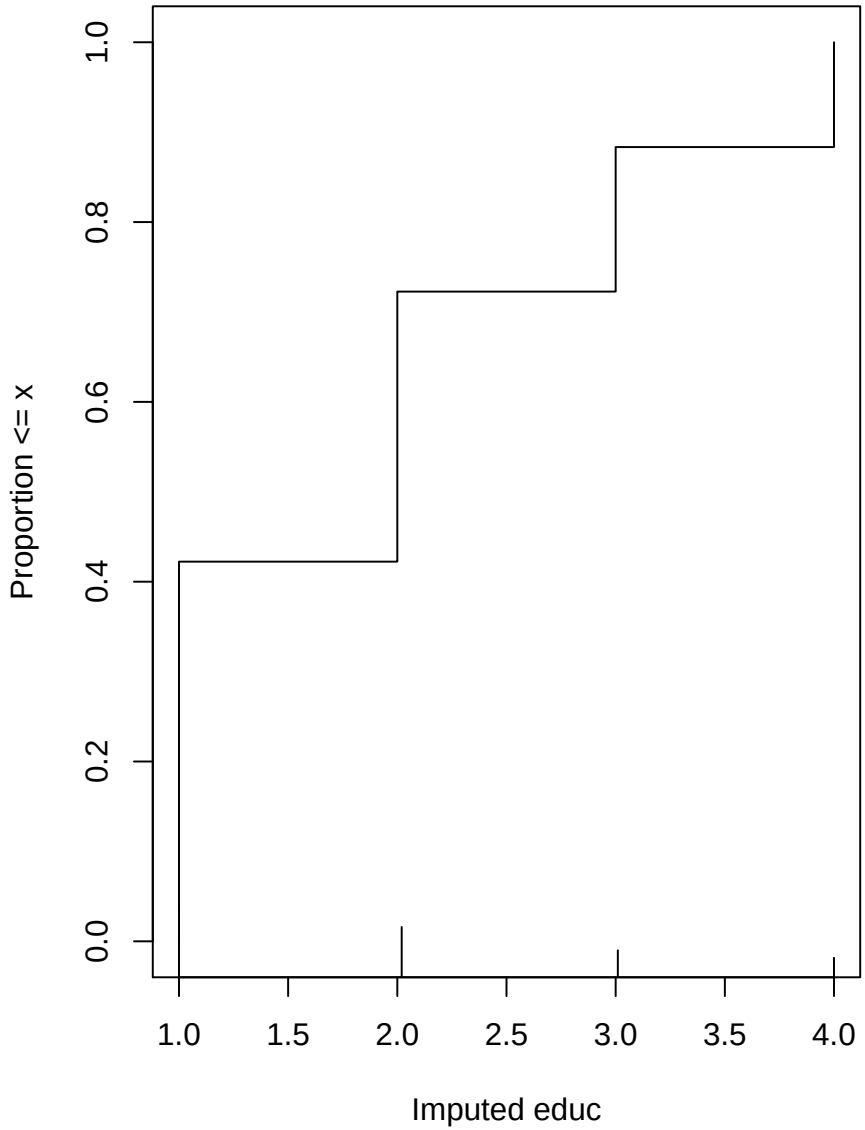
```
Iteration 1 Iteration 2 Iteration 3 Iteration 4 Iteration 5 It
```

Imputation Results (edited)

```
fit_imp

Multiple Imputation using Bootstrap and PMM

aregImpute(formula = ~chd10 + glucose + smoker + sbp + educ,
    data = fram, n.impute = 50, nk = c(0, 3:5),
    tlinear = FALSE, B = 10)

n: 4238     p: 5    Imputations: 50     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
   0.032   0.028
```

Plots of Multiply Imputed Values





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 = 50)
```

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

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

Fitting modC (modA with multiple imputation)

Results summarized on next few slides. . .

```
modC <-
    fit.mult.impute(chd10 ~ glucose + smoker + sbp + educ,
                    fitter = lrm, xtrans = fit_imp,
                    data = fram, x = TRUE, y = TRUE)
```

```
Variance Inflation Factors Due to Imputation:

     Intercept          glucose           smoker              sbp
          1.02             1.08             1.00             1.00
  educ=HS grad educ=Some Coll educ=Coll grad
          1.02             1.02             1.02

Rate of Missing Information:

     Intercept          glucose           smoker              sbp
```

modC instant output

```
Variance Inflation Factors Due to Imputation:

     Intercept          glucose           smoker              sbp
          1.02             1.08             1.00             1.00
  educ=HS grad educ=Some Coll educ=Coll grad
          1.02             1.02             1.02

Rate of Missing Information:

     Intercept          glucose           smoker              sbp
          0.02             0.08             0.00             0.00
  educ=HS grad educ=Some Coll educ=Coll grad
          0.02             0.02             0.02
```

`modC` coefficients, compared to `A` and `B`

- These still are changes in the log odds ratio for `chd10`.
- Exponentiate to get odds ratios.

Term	Model C	Model B	Model A
Intercept	-5.548 (0.308)	-5.565 (0.307)	-5.562 (0.322)
glucose	0.008 (0.002)	0.009 (0.002)	0.008 (0.002)
smoker	0.318 (0.090)	0.321 (0.090)	0.313 (0.096)
sbp	0.023 (0.002)	0.023 (0.002)	0.024 (0.002)
educ=HS grad	-0.452 (0.111)	-0.471 (0.110)	-0.467 (0.116)
educ=Some Coll	-0.299 (0.134)	-0.306 (0.134)	-0.392 (0.142)
educ=Coll grad	-0.085 (0.148)	-0.082 (0.147)	-0.136 (0.155)

modC summaries, compared to modB and modA

Summary	modC	modB	modA
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	0.676	0.677	0.682
Dxy	0.353	0.354	0.363

Validate modC Summary Statistics

```
set.seed(2020)
validate(modC, B = 50)
```

Edited Results to fit on the screen...

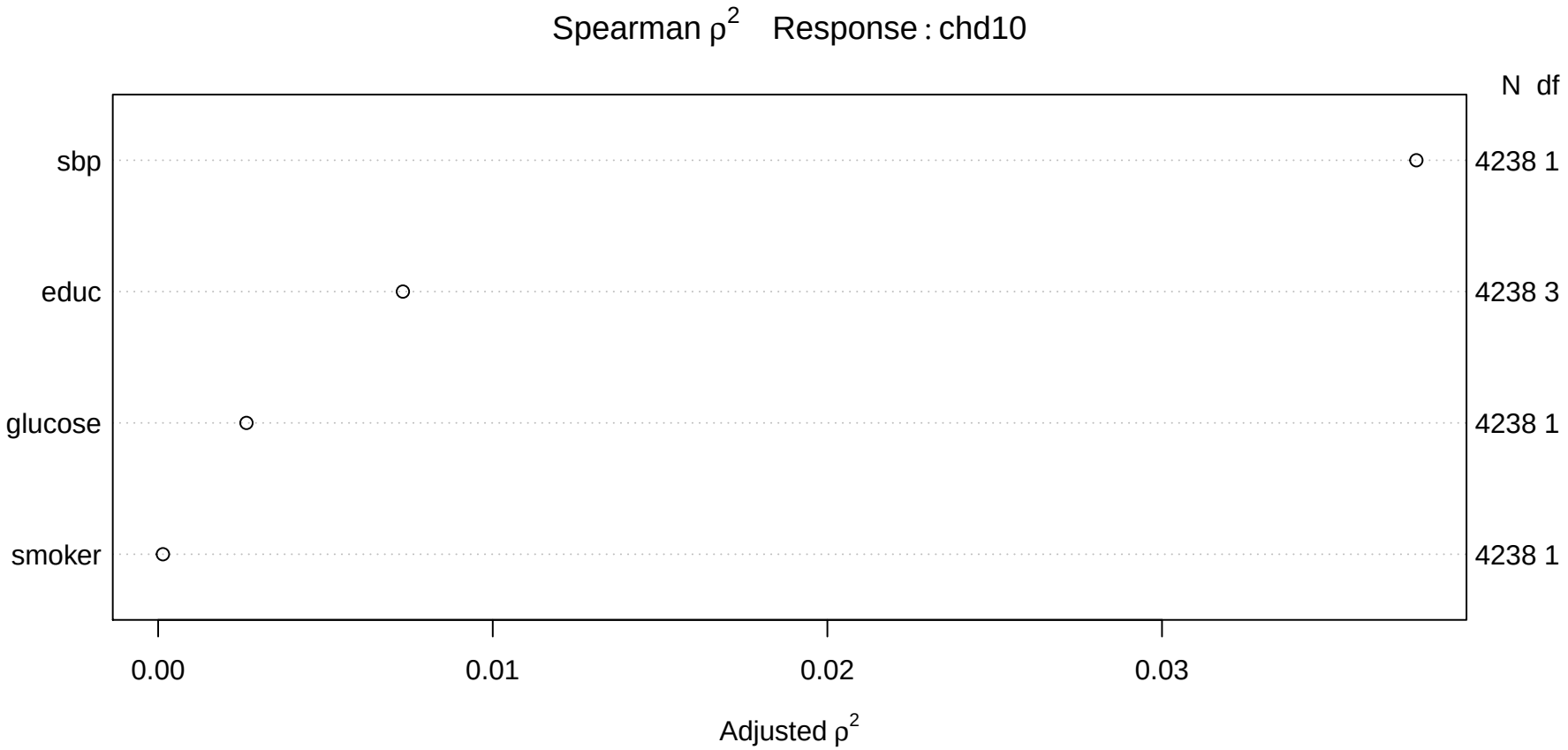
	index.orig	training	test	optimism	index.corrected	n
Dxy	0.3510	0.3488	0.3465	0.0024	0.3486	50
R2	0.0947	0.0926	0.0909	0.0017	0.0930	50
B	0.1208	0.1209	0.1211	-0.0002	0.1210	50

Again, remember that $C = 0.5 + \frac{Dxy}{2}$.

Adding some 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. . .

modD incorporating single imputation

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

modD <- lrm(chd10 ~ rcs(sbp, 5) + pol(glucose, 2) +
                    smoker + educ + educ %ia% sbp,
              data = fram_sh, x = TRUE, y = TRUE)
```

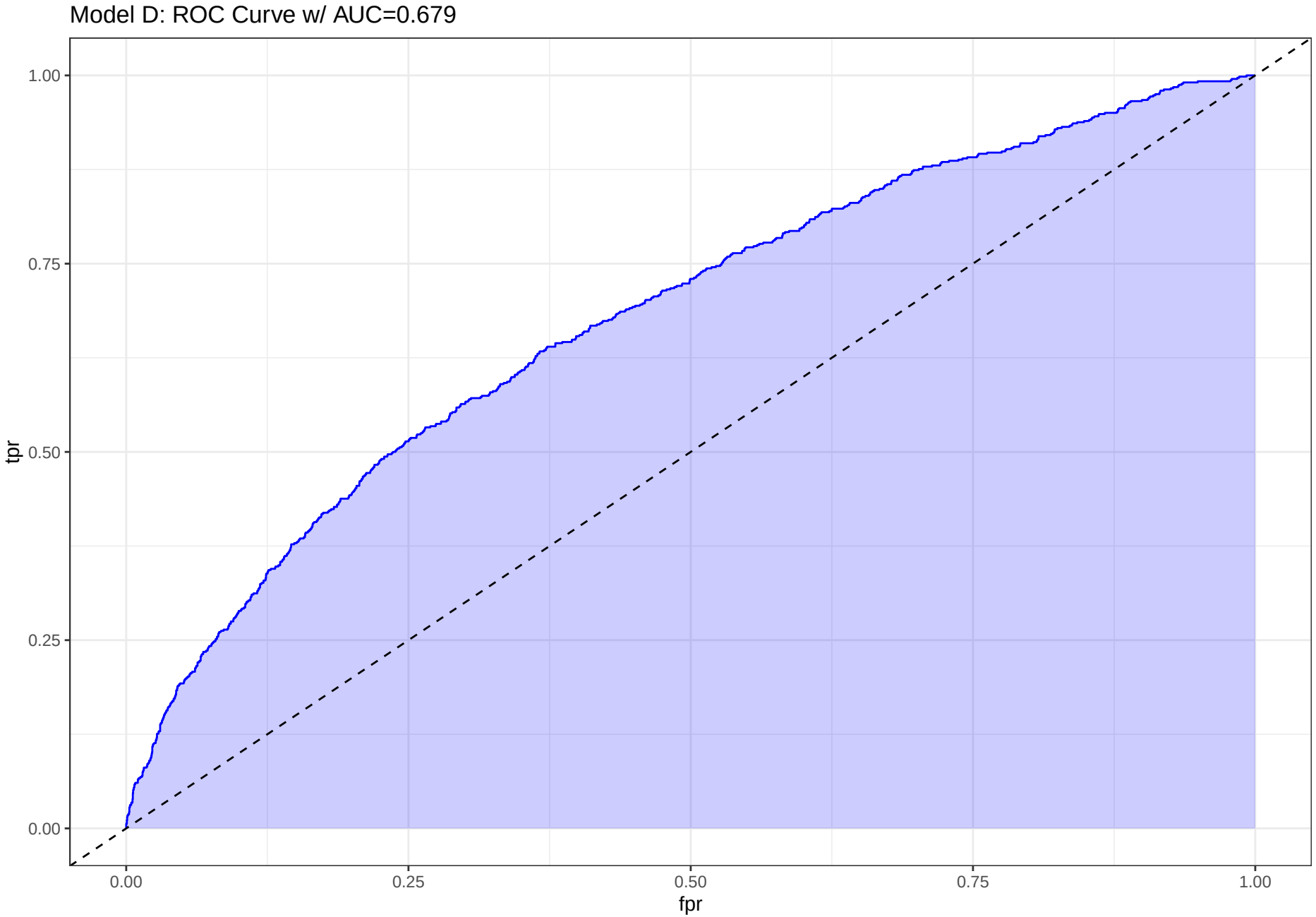
modD Coefficients

	Coef	S.E.	Wald Z	Pr(>|Z|)
Intercept	-3.2491	2.1109	-1.54	0.1238
sbp	0.0035	0.0190	0.18	0.8544
sbp'	0.1745	0.1836	0.95	0.3420
sbp''	-0.5032	0.6399	-0.79	0.4316
sbp'''	0.3642	0.6487	0.56	0.5745
glucose	0.0058	0.0053	1.09	0.2769
glucose^2	0.0000	0.0000	0.57	0.5689
smoker	0.3235	0.0902	3.59	0.0003
educ=HS grad	-0.4905	0.6353	-0.77	0.4401
educ=Some Coll	-1.4647	0.8032	-1.82	0.0682
educ=Coll grad	-1.1596	0.9327	-1.24	0.2137
educ=HS grad * sbp	0.0001	0.0045	0.02	0.9847
educ=Some Coll * sbp	0.0084	0.0057	1.47	0.1406
educ=Coll grad * sbp	0.0079	0.0067	1.18	0.2399

`modD` summaries vs. `modB` without non-linear pieces

Summary	`modD`	`modB`
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	0.358	0.354

ROC Curve for our modD

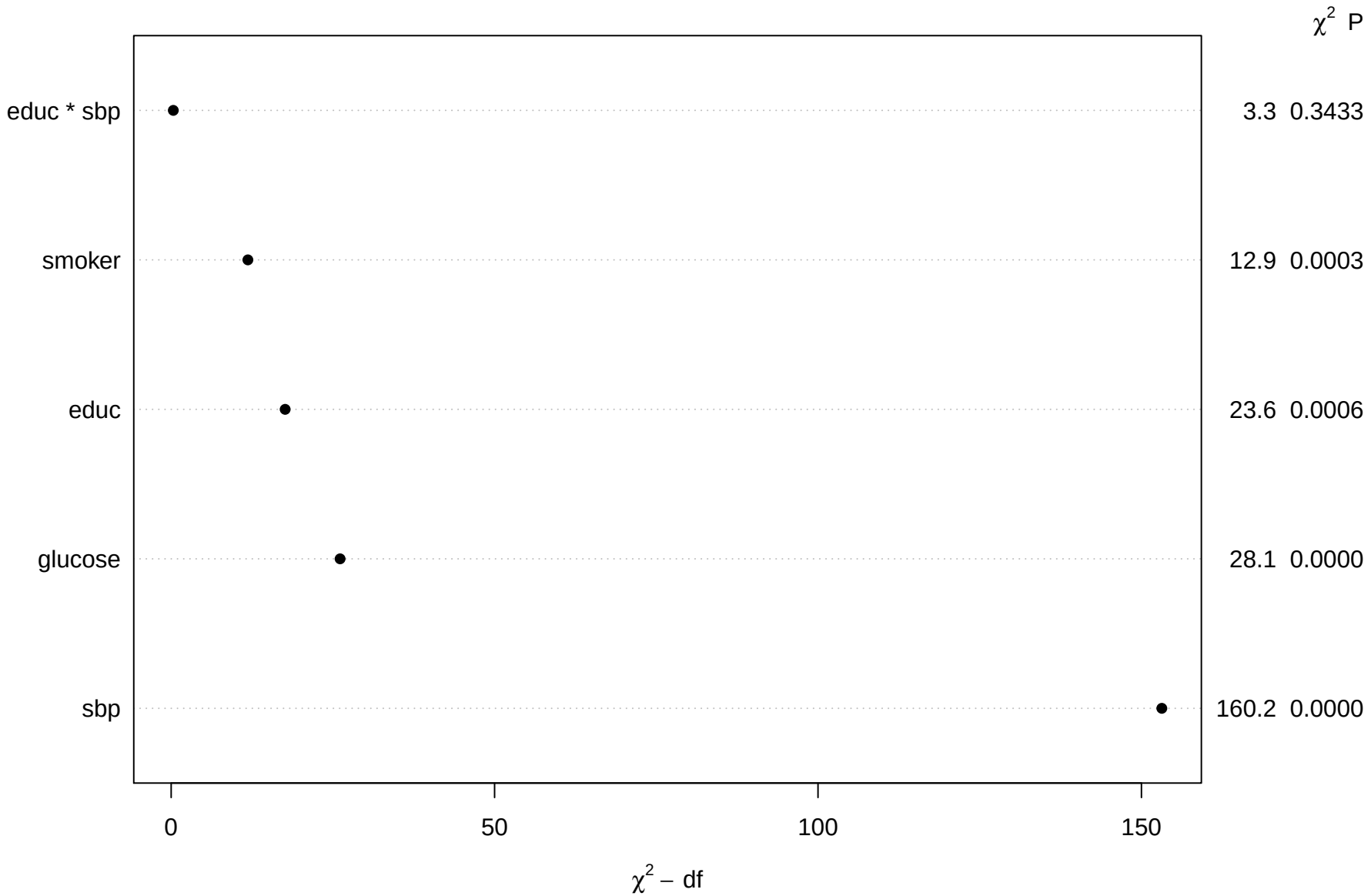


anova(modD) edited to fit on the screen

Wald Statistics Response: chd10

Factor	Chi-Square	d.f.	P
sbp	160.15	7	<.0001
All Interactions	3.33	3	0.3433
Nonlinear	3.02	3	0.3883
glucose	28.12	2	<.0001
Nonlinear	0.32	1	0.5689
smoker	12.86	1	0.0003
educ	23.62	6	0.0006
All Interactions	3.33	3	0.3433
educ * sbp	3.33	3	0.3433
TOTAL NONLINEAR	3.28	4	0.5119
TOTAL NONLINEAR + INTERACTION	7.28	7	0.4001
TOTAL	227.47	13	<.0001

plot(anova(modD)) (uses 13 df)



Validate modD Summary Statistics

```
set.seed(2020)
validate(modD, B = 50)
```

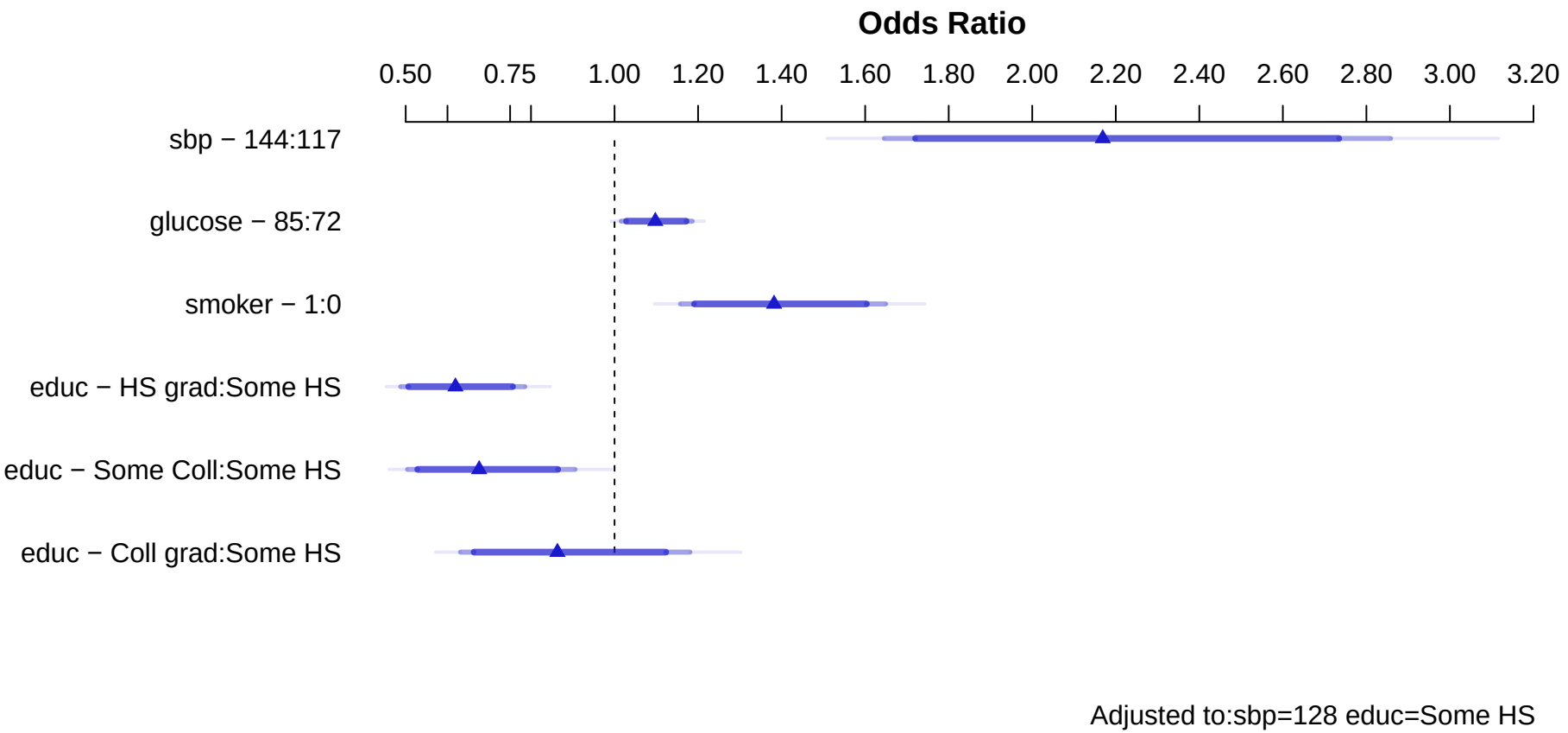
Edited Results to fit on the screen...

	index.orig	training	test	optimism	index.corrected	n
Dxy	0.3576	0.3605	0.3507	0.0098	0.3478	50
R2	0.0982	0.1008	0.0932	0.0076	0.0907	50
B	0.1203	0.1201	0.1208	-0.0007	0.1210	50

Again, remember that $C = 0.5 + \frac{Dxy}{2}$.

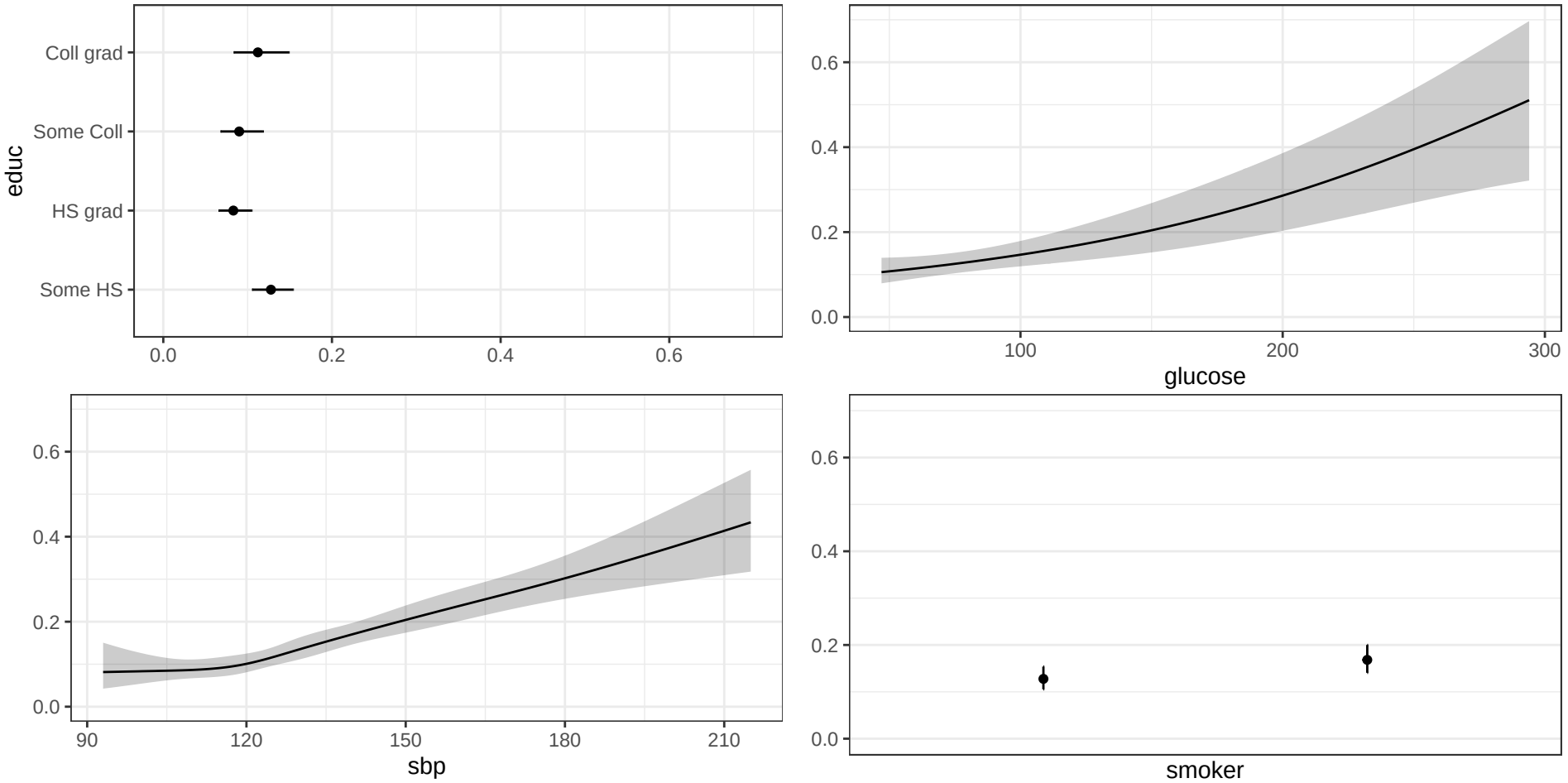
Plot of Effects using modD

```
plot(summary(modD))
```



Predict results for modD

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



Nomogram for modD

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