# 431 Class 23

Thomas E. Love, Ph.D.

2022-12-06

# Today's Agenda

1. What exactly is R doing if you ignore missing values when fitting models?
    - What does `type.convert()` do?
    - `na.omit` vs. `na.exclude` vs. `na.delete`
2. Use multiple imputation to deal with missing data in fitting a linear regression with `lm` using the `mice` package.

(MICE = Multiple Imputation through Chained Equations)

# Today’s Packages

```
library(magrittr); library(knitr); library(kableExtra)
library(janitor); library(naniar); library(broom)
library(car); library(GGally)
library(mice); library(mitml)
  # mice = multiple imputation through chained equations
library(tidyverse)

theme_set(theme_bw())
```

# What happens if you fit a regression model without doing anything at all about missing data?

# What happens if you ignore NAs?

Let’s open a small, simulated data set with 100 subjects and some missing values.

```r
sim1 <- read_csv("c23/data/c23_sim1.csv") |>
    type.convert(as.is = FALSE, na.strings = "NA")

head(sim1)
```

```
# A tibble: 6 × 6
  subject out_q out_b pred1 pred2 pred3
  <fct>   <dbl> <fct> <dbl> <dbl> <fct>
1 S001     81.1 Yes     8.8  20.5 Middle
2 S002    105.  No      7.1  24.9 High
3 S003     NA   <NA>    9.9  17.4 Middle
4 S004     NA   No      8.9  31.8 <NA>
5 S005     75.9 <NA>   NA    22   High
6 S006     79.8 No      9.7  NA   <NA>
```

# What does `type.convert()` do?

Tries to convert each column (individually) to either logical, integer, numeric, complex or (if a character vector) to factor.

- The first type (from that list) that can accept all non-missing values is chosen.
- If all values are missing, the column is converted to logical.
- Columns containing just `F`, `T`, `FALSE`, `TRUE` or `NA` values are made into logical.
- Use the `na.strings` parameter to add missing strings (default = `"NA"`)
- `as.is = FALSE` converts characters to factors. `as.is = TRUE` is the default.

# Our `sim1` data

| Variable | Description |
|---|---|
| `subject` | Subject identifier |
| `out_q` | Quantitative outcome |
| `out_b` | Binary outcome with levels Yes, No |
| `pred1` | Predictor 1 (quantitative) |
| `pred2` | Predictor 2 (also quantitative) |
| `pred3` | Predictor 3 (categories are Low, Middle, High) |

- Clean up the factors?

# Cleaning up subject and pred3

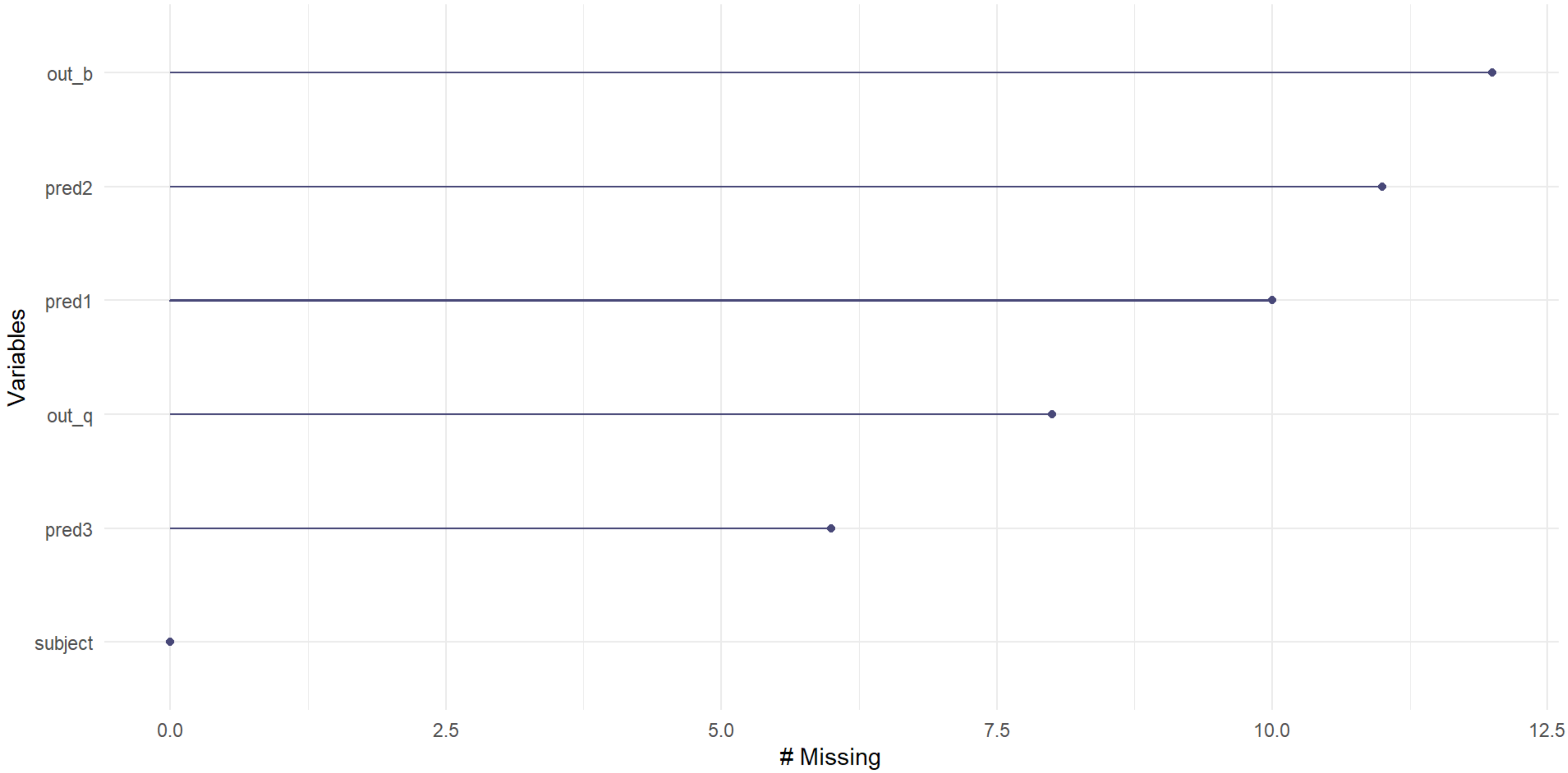
```
sim1 <- sim1 |>
    mutate(subject = as.character(subject),
           pred3 = fct_relevel(pred3, "Low", "Middle"))

sim1 |> tabyl(pred3, out_b)
```

```
  pred3 No Yes NA_
    Low 10  12   4
 Middle 12  17   4
   High 16  15   4
   <NA>  4   2   0
```

# How much missingness do we have?

```
gg_miss_var(sim1)
```



# How much missingness do we have?

```
miss_var_summary(sim1)
```

```
# A tibble: 6 × 3
  variable n_miss pct_miss
  <chr>     <int>    <dbl>
1 out_b        12       12
2 pred2        11       11
3 pred1        10       10
4 out_q         8        8
5 pred3         6        6
6 subject       0        0
```

```
n_miss(sim1)
```

```
[1] 47
```

# How much missingness do we have?

```
prop_complete_case(sim1)
```

```
[1] 0.65
```

```
miss_case_table(sim1)
```

```
# A tibble: 4 × 3
  n_miss_in_case n_cases pct_cases
           <int>   <int>     <dbl>
1              0      65        65
2              1      25        25
3              2       8         8
4              3       2         2
```

# Suppose we run a linear regression

## without dealing with the missing data, so that we run:

```
mod1 <- lm(out_q ~ pred1 + pred2 + pred3, data = sim1)
summary(mod1)
```

```
Call:
lm(formula = out_q ~ pred1 + pred2 + pred3, data = sim1)

Residuals:
    Min      1Q  Median      3Q     Max 
-39.164 -13.900   2.419  15.541  34.156 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 105.2070    18.6185   5.651 3.82e-07 ***
pred1        -0.8361     1.3010  -0.643    0.523    
pred2         0.2611     0.4614   0.566    0.573    
pred3Middle  -1.3498     5.6802  -0.238    0.813    
pred3High    -2.7443     5.5427  -0.495    0.622    
```

## How can we tell how many observations will be used?

# What happens when we run a regression model?

```r
mod1 <- lm(out_q ~ pred1 + pred2 + pred3, data = sim1)

anova(mod1)
```

```
Analysis of Variance Table

Response: out_q
          Df  Sum Sq Mean Sq F value Pr(>F)
pred1      1   209.8  209.81  0.5976 0.4423
pred2      1   132.1  132.14  0.3763 0.5417
pred3      2    86.5   43.24  0.1231 0.8843
Residuals 65 22821.9  351.11
```

- How many observations were used to fit this model?

# Another way to see this

```
glance(mod1) |> select(1:6)
```

```
# A tibble: 1 × 6
  r.squared adj.r.squared sigma statistic p.value    df
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>
1    0.0184       -0.0420  18.7     0.305   0.874     4
```

```
glance(mod1) |> select(7:12)
```

```
# A tibble: 1 × 6
  logLik   AIC   BIC deviance df.residual  nobs
   <dbl> <dbl> <dbl>    <dbl>       <int> <int>
1  -302.  616.  629.   22822.          65    70
```

# How could we have known this would be 70, in advance?

```r
sim1 |> select(out_q, pred1, pred2, pred3) |>
    miss_case_table()
```

```
# A tibble: 3 × 3
  n_miss_in_case n_cases pct_cases
           <int>   <int>     <dbl>
1              0      70        70
2              1      25        25
3              2       5         5
```

# Which observations were not used?

```
summary(mod1)$na.action
```

```
 3  4  5  6 13 16 19 26 27 29 30 34 39 48 51 56 62 66 67 68 72 75 81 83 86 89 
 3  4  5  6 13 16 19 26 27 29 30 34 39 48 51 56 62 66 67 68 72 75 81 83 86 89 
93 94 96 97 
93 94 96 97 
attr(,"class")
[1] "omit"
```

- A potentially more useful `na.action` setting in `lm` is `na.exclude` which pads out predicted values and residuals with NAs instead of omitting the 30 observations listed above.

```
lm(out_q ~ pred1 + pred2 + pred3,
      data = sim1, na.action = na.exclude)
```

# Predictions from `mod1` with `na.omit` and `na.exclude`

```r
mod1 <- lm(out_q ~ pred1 + pred2 + pred3, data = sim1)
               ## note: by default na.action = na.omit here
head(predict(mod1))
```

```
        1         2         7         8         9        10 
101.85279 103.02874  98.14391  96.57037 101.49208 101.01744 
```

```r
mod1_e <- lm(out_q ~ pred1 + pred2 + pred3, data = sim1,
             na.action = na.exclude)
head(predict(mod1_e))
```

```
       1        2        3        4        5        6 
101.8528 103.0287       NA       NA       NA       NA 
```

# Multiple Imputation: Potential and Pitfalls

# Sterne et al. 2009 *BMJ*

Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls

> In this article, we review the reasons why missing data may lead to bias and loss of information in epidemiological and clinical research. We discuss the circumstances in which multiple imputation may help by reducing bias or increasing precision, as well as describing potential pitfalls in its application. Finally, we describe the recent use and reporting of analyses using multiple imputation in general medical journals, and suggest guidelines for the conduct and reporting of such analyses.

- https://www.bmj.com/content/338/bmj.b2393

**Note**: The next 7 slides are derived from Sterne et al.

# An Example from Sterne et al.

Consider, for example, a study investigating the association of systolic blood pressure with the risk of subsequent coronary heart disease, in which data on systolic blood pressure are missing for some people.

The probability that systolic blood pressure is missing is likely to:

- decrease with age (doctors are more likely to measure it in older people),
- decrease with increasing body mass index, and
- decrease with history of smoking (doctors are more likely to measure it in people with heart disease risk factors or comorbidities).

If we assume that data are missing at random and that we have systolic blood pressure data on a representative sample of individuals within strata of age, smoking, body mass index, and coronary heart disease, then we can use multiple imputation to estimate the overall association between systolic blood pressure and coronary heart disease.

# Missing Data Mechanisms

- **Missing completely at random** There are no systematic differences between the missing values and the observed values.
    - For example, blood pressure measurements may be missing because of breakdown of an automatic sphygmomanometer.
- **Missing at random** Any systematic difference between the missing and observed values can be explained by other observed data.
    - For example, missing BP measurements may be lower than measured BPs but only because younger people more often have a missing BP.
- **Missing not at random** Even after the observed data are taken into account, systematic differences remain between the missing values and the observed values.
    - For example, people with high BP may be more likely to have headaches that cause them to miss clinic appointments.

“Missing at random” is an **assumption** that justifies the analysis, and is not a property of the data.

# Trouble: Data missing not at random

Sometimes, it is impossible to account for systematic differences between missing and observed values using the available data.

- In such (MNAR) cases, multiple imputation may give misleading results.
  - Those results can be either more or less misleading than a complete case analysis.
- For example, consider a study investigating predictors of depression.
  - If individuals are more likely to miss appointments because they are depressed on the day of the appointment, then it may be impossible to make the MAR assumption plausible, even if a large number of variables is included in the imputation model.

Where complete cases and multiple imputation analyses give different results, the analyst should attempt to understand why, and this should be reported in publications.

# What if the data are MCAR?

If we assume data are MAR, then unbiased and statistically more powerful analyses (compared with analyses based on complete cases) can generally be done by including individuals with incomplete data.

There are circumstances in which analyses of **complete cases** will not lead to bias.

- Missing data in predictor variables do not cause bias in analyses of complete cases if the reasons for the missing data are unrelated to the outcome.
  - In such cases, imputing missing data may lessen the loss of precision and power resulting from exclusion of individuals with incomplete predictor variables but are not required in order to avoid bias.

# Stages of Multiple Imputation (1 of 2)

> Multiple imputation … aims to allow for the uncertainty about the missing data by creating several different plausible imputed data sets and appropriately combining results obtained from each of them.

The first stage is to create multiple copies of the dataset, with the missing values replaced by imputed values.

- The imputation procedure must fully account for all uncertainty in predicting the missing values by injecting appropriate variability into the multiple imputed values; we can never know the true values of the missing data.

Note that single Imputation of missing values usually causes standard errors to be too small, since it fails to account for the fact that we are uncertain about the missing values.

# Stages of Multiple Imputation (2 of 2)

The second stage is to use standard statistical methods to fit the model of interest to each of the imputed datasets.

- Estimated associations in each of the imputed datasets will differ because of the variation introduced in the imputation of the missing values, and they are only useful when averaged together to give overall estimated associations.
- Standard errors are calculated using Rubin’s rules, which take account of the variability in results between the imputed datasets, reflecting the uncertainty associated with the missing values.
- Valid inferences are obtained because we are averaging over the distribution of the missing data given the observed data.

# Comparing Two Linear Models including Multiple Imputation

# Framingham data

```r
fram_raw <- read_csv("c23/data/framingham.csv", show_col_types = FALSE) |>
    clean_names()

dim(fram_raw)
```

```
[1] 4238   17
```

```r
n_miss(fram_raw)
```

```
[1] 645
```

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

# Codebook for Today

| Variable | Description |
|---:|---|
| `educ` | four-level factor: educational attainment |
| `smoker` | 1 = current smoker at examination time, else 0 |
| `sbp` | systolic blood pressure (mm Hg) |
| `obese` | 1 if subject’s `bmi` is 30 or higher, else 0 |
| `glucose` | blood glucose level in mg/dl |

- The variables describe 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.

# fram_sub Tibble for Today

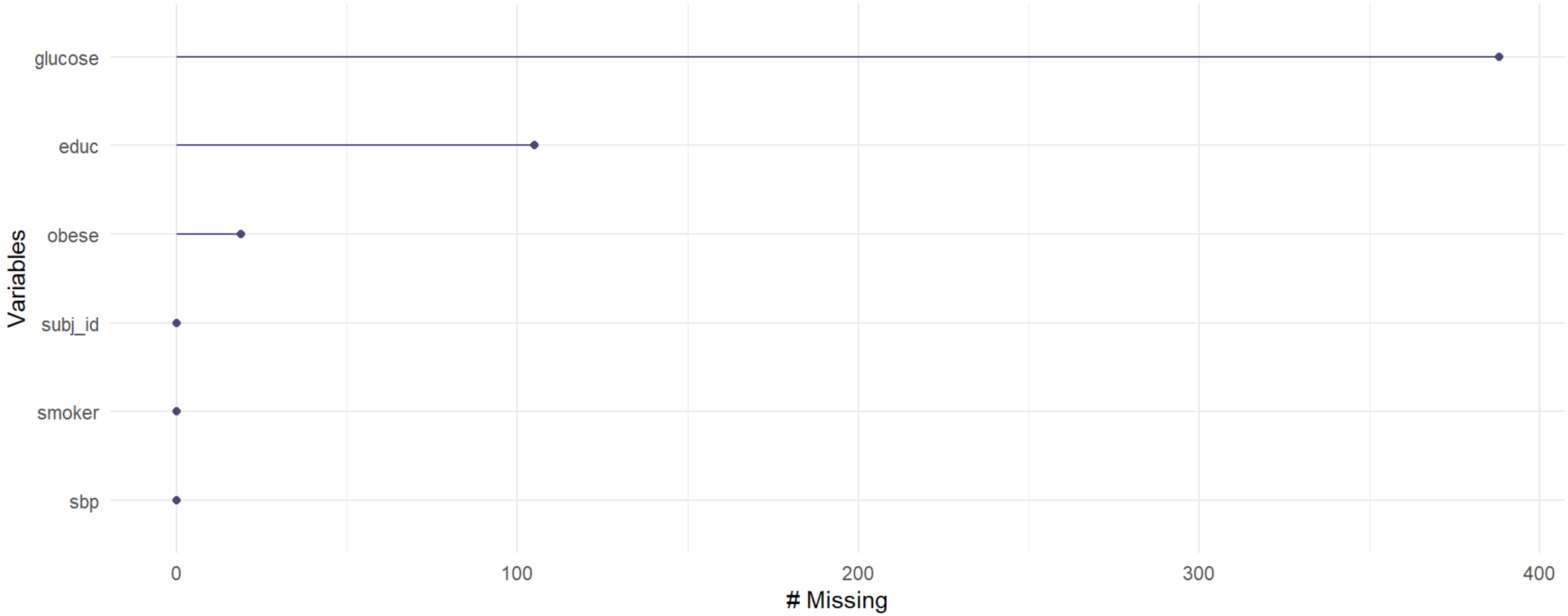
```
fram_sub <- fram_raw |>
    mutate(educ = fct_recode(factor(education),
                              "Some HS" = "1",
                              "HS grad" = "2",
                              "Some Coll" = "3",
                              "Coll grad" = "4")) |>
    mutate(obese = as.numeric(bmi >= 30)) |>
    rename(smoker = "current_smoker",
           sbp = "sys_bp") |>
    mutate(subj_id = as.character(subj_id)) |>
    select(sbp, educ, smoker, obese, glucose, subj_id)

dim(fram_sub)
```

```
[1] 4238    6
```

# Which variables are missing data?

```
gg_miss_var(fram_sub)
```

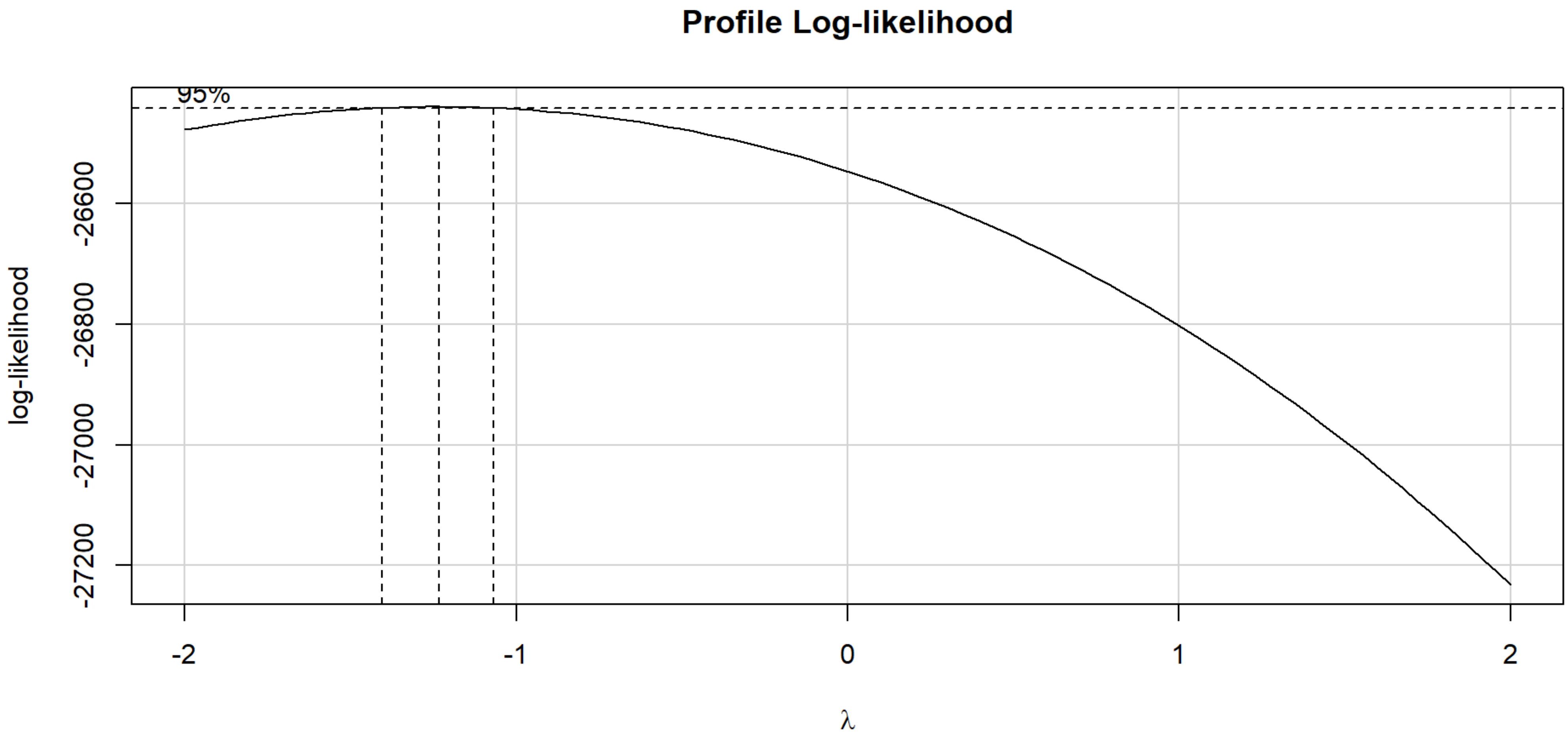


# Today’s Goal

Use linear regression to predict `sbp` using two different models, in each case accounting for missingness via multiple imputation, where the predictors of interest are `glucose`, `obese`, `educ`, and `smoker`.

# Consider a transformation?

```
with(fram_sub, car::boxCox(sbp ~ glucose + obese + educ + smoker))
```



# Create a new outcome variable

```r
fram_sub <- fram_sub |>
  mutate(inv_sbp = 1000 / sbp)

summary(1/fram_sub$sbp)
```

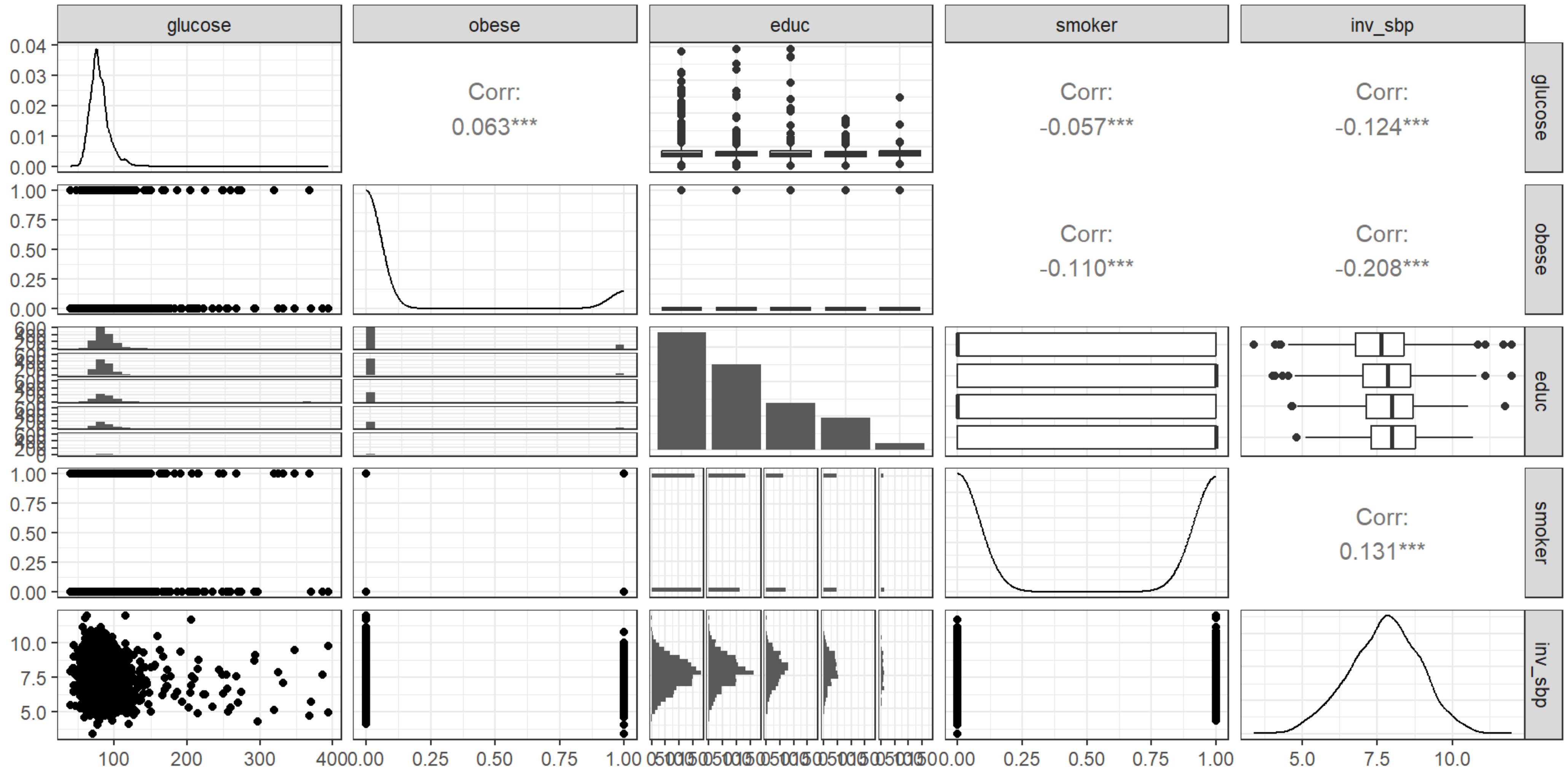
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
0.003390 0.006944 0.007812 0.007746 0.008547 0.011976 
```

```r
summary(fram_sub$inv_sbp)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  3.390   6.944   7.812   7.746   8.547  11.976 
```

# Scatterplot Matrix (no imputation)

```
ggpairs(fram_sub |> select(glucose, obese, educ, smoker, inv_sbp))
```



# Track missingness with shadow

```r
fram_sub_sh <- bind_shadow(fram_sub)

head(fram_sub_sh)
```

```
# A tibble: 6 × 14
    sbp educ      smoker obese glucose subj_id inv_sbp sbp_NA educ_NA
smoker_NA
  <dbl> <fct>      <dbl> <dbl>   <dbl> <chr>     <dbl> <fct>  <fct>   <fct>
1  106  Coll grad      0     0      77 1          9.43 !NA    !NA     !NA
2  121  HS grad        0     0      76 2          8.26 !NA    !NA     !NA
3  128. Some HS        1     0      70 3          7.84 !NA    !NA     !NA
4  150  Some Coll      1     0     103 4          6.67 !NA    !NA     !NA
5  130  Some Coll      1     0      85 5          7.69 !NA    !NA     !NA
6  180  HS grad        0     1      99 6          5.56 !NA    !NA     !NA
# … with 4 more variables: obese_NA <fct>, glucose_NA <fct>, subj_id_NA <fct>,
#   inv_sbp_NA <fct>
```

# Our Two Models

Model 2: predict 1000/`sbp` using `glucose` and `obese`.

Model 4: predict 1000/`sbp` using `glucose`, `obese`, `educ`, and `smoker`.

# Model 2 (CC): 2 predictors

Suppose we ignore the missingness and just run the model on the data with complete information on `inv_sbp`, `glucose` and `obese`.

```r
m2_cc <- with(fram_sub_sh, lm(inv_sbp ~ glucose + obese))

tidy(m2_cc, conf.int = TRUE, conf.level = 0.95) |> select(-statistic) |>
    kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | p.value | conf.low | conf.high |
|---|---:|---:|---:|---:|---:|
| (Intercept) | 8.259 | 0.066 | 0 | 8.129 | 8.389 |
| glucose | -0.005 | 0.001 | 0 | -0.007 | -0.004 |
| obese | -0.719 | 0.056 | 0 | -0.828 | -0.610 |

# Edited Summary of Model 2 (CC)

```r
summary(m2_cc)    ## we'll just look at the bottom
```

```
Residual standard error: 1.14 on 3833 degrees of freedom
  (402 observations deleted due to missingness)
Multiple R-squared:  0.05531,   Adjusted R-squared:  0.05481 
F-statistic: 112.2 on 2 and 3833 DF,  p-value: < 2.2e-16
```

```r
glance(m2_cc) |>
    select(nobs, r.squared, adj.r.squared, AIC, BIC) |>
    kable(digits = c(0, 4, 4, 0, 0)) |> kable_styling(font_size = 28)
```

| nobs | r.squared | adj.r.squared | AIC | BIC |
|---:|---:|---:|---:|---:|
| 3836 | 0.0553 | 0.0548 | 11894 | 11919 |

# Model 4 (CC): 4 predictors

```r
m4_cc <- lm(inv_sbp ~ glucose + obese + smoker + educ, data = fram_sub_sh)

tidy(m4_cc, conf.int = TRUE) |> select(-statistic) |>
    kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | p.value | conf.low | conf.high |
|---|---:|---:|---:|---:|---:|
| (Intercept) | 7.967 | 0.074 | 0 | 7.822 | 8.111 |
| glucose | -0.005 | 0.001 | 0 | -0.006 | -0.003 |
| obese | -0.650 | 0.057 | 0 | -0.761 | -0.539 |
| smoker | 0.253 | 0.037 | 0 | 0.180 | 0.325 |
| educHS grad | 0.196 | 0.044 | 0 | 0.109 | 0.283 |
| educSome Coll | 0.251 | 0.054 | 0 | 0.146 | 0.357 |
| educColl grad | 0.317 | 0.062 | 0 | 0.196 | 0.438 |

# Edited Summary of Model 4 (CC)

```r
summary(m4_cc)              ## we'll just look at the bottom
```

```
Residual standard error: 1.126 on 3733 degrees of freedom
  (498 observations deleted due to missingness)
Multiple R-squared:  0.07919,	Adjusted R-squared:  0.07771 
F-statistic:  53.5 on 6 and 3733 DF,  p-value: < 2.2e-16
```

```r
glance(m4_cc) |>
    select(nobs, r.squared, adj.r.squared, AIC, BIC) |>
    kable(digits = c(0, 4, 4, 0, 0)) |> kable_styling(font_size = 28)
```

| nobs | r.squared | adj.r.squared | AIC | BIC |
|---:|---:|---:|---:|---:|
| 3740 | 0.0792 | 0.0777 | 11513 | 11563 |

# Variables used in our models 2 and 4

```
miss_var_summary(fram_sub)
```

```
# A tibble: 7 × 3
  variable n_miss pct_miss
  <chr>     <int>    <dbl>
1 glucose     388    9.16
2 educ        105    2.48
3 obese        19    0.448
4 sbp           0    0
5 smoker        0    0
6 subj_id       0    0
7 inv_sbp       0    0
```

- Are we missing data on our outcome for these models?

# Create multiple imputations

How many subjects have complete / missing data that affect this model?

```
pct_complete_case(fram_sub)
```

```
[1] 88.24917
```

```
pct_miss_case(fram_sub)
```

```
[1] 11.75083
```

## Let's create 15 imputed data sets. (Why 15?)

```
set.seed(431431)
fram_mice24 <- mice(fram_sub, m = 15, printFlag = FALSE)
```

- Using `printFlag = FALSE` eliminates a lot of unnecessary (and not particularly informative) output.

# Summary of Imputation Process

```r
summary(fram_mice24)
```

```
Class: mids
Number of multiple imputations:  15 
Imputation methods:
      sbp      educ    smoker     obese   glucose   subj_id   inv_sbp 
       "" "polyreg"        ""     "pmm"     "pmm"        ""        "" 
PredictorMatrix:
        sbp educ smoker obese glucose subj_id inv_sbp
sbp       0    1      1     1       1       0       1
educ      1    0      1     1       1       0       1
smoker    1    1      0     1       1       0       1
obese     1    1      1     0       1       0       1
glucose   1    1      1     1       0       0       1
subj_id   1    1      1     1       1       0       1
Number of logged events:  1 
  it im dep     meth     out
1  0  0     constant subj_id
```

- See Heymans and Eekhout sections 4.6 - 4.14 for more information.

# Imputation Options within `mice`

Default methods include:

- `pmm` predictive mean matching (default choice for quantitative variables)
- `logreg` logistic regression (default for binary categorical variables)
- `polyreg` polytomous logistic regression (for nominal multi-categorical variables)
- `polr` proportional odds logistic regression (for ordinal categories)

but there are `cart` methods and many others available, too.

# What should we include in an imputation model?

1. If things you are imputing are not Normally distributed, this can pose special challenges, and either a transformation or choosing an imputation method which is robust to these concerns is helpful.
2. Include the outcome when imputing predictors. It causes you to conclude the relationship is weaker than it actually is, if you don’t.
3. The MAR assumption may only be reasonable when a certain variable is included in the model.
    - As a result, it’s usually a good idea to include as wide a range of variables in imputation models as possible. The concerns we’d have about parsimony in outcome models don’t apply here.

# Store one (or more) of the imputed data sets

This will store the fifth imputed data set in `imp_5`.

```r
imp_5 <- complete(fram_mice24, 5) |> tibble()

dim(imp_5)
```

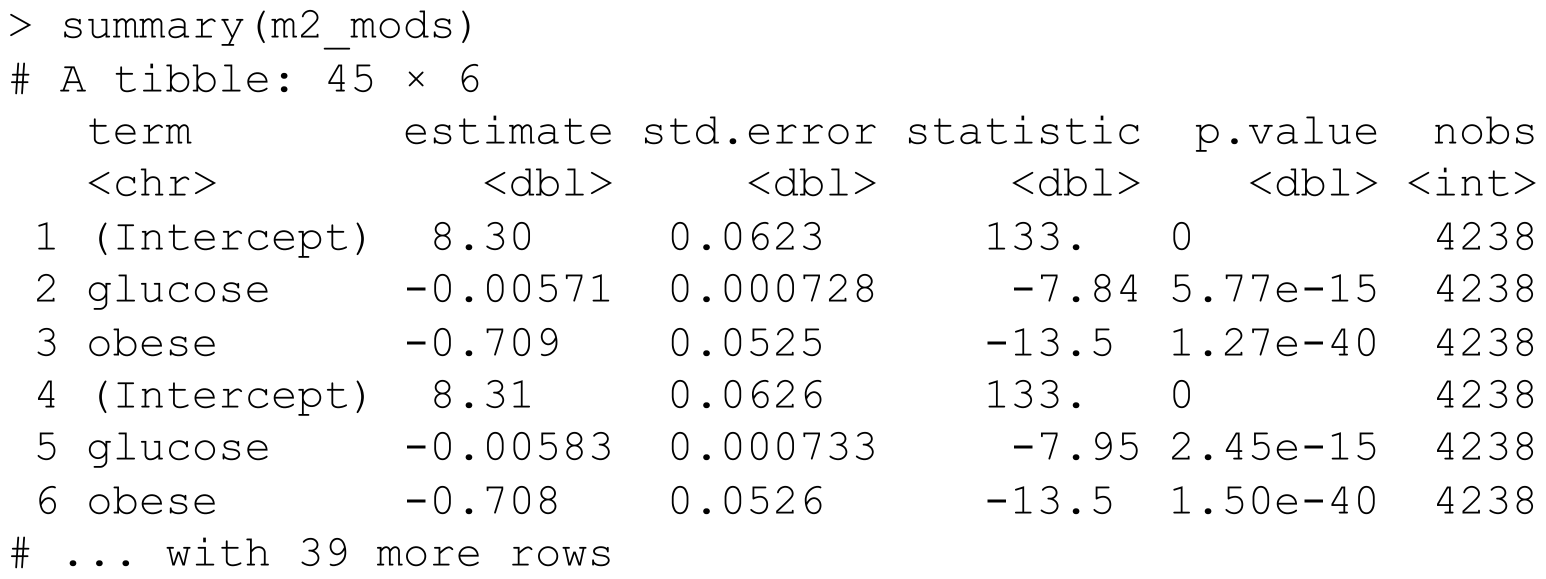
```
[1] 4238    7
```

```r
n_miss(imp_5)
```

```
[1] 0
```

# Run Model 2 on each imputed data frame

```
m2_mods <- with(fram_mice24, lm(inv_sbp ~ glucose + obese))
```



```
> summary(m2_mods)
# A tibble: 45 × 6
   term        estimate std.error statistic  p.value  nobs
   <chr>          <dbl>     <dbl>     <dbl>    <dbl> <int>
 1 (Intercept)  8.30     0.0623      133.   0            4238
 2 glucose     -0.00571  0.000728     -7.84 5.77e-15  4238
 3 obese       -0.709    0.0525      -13.5  1.27e-40  4238
 4 (Intercept)  8.31     0.0626      133.   0            4238
 5 glucose     -0.00583  0.000733     -7.95 2.45e-15  4238
 6 obese       -0.708    0.0526      -13.5  1.50e-40  4238
# ... with 39 more rows
```

- 3 coefficients in each model, times 15 imputations = 45 rows.

# More detailed regression results?

Consider working with the analysis done on the 4th imputed data set (of the 15 created)…

```r
m2_a4 <- m2_mods$analyses[[4]]
tidy(m2_a4) |> kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | statistic | p.value |
|:---|---:|---:|---:|---:|
| (Intercept) | 8.270 | 0.063 | 132.105 | 0 |
| glucose | -0.005 | 0.001 | -7.191 | 0 |
| obese | -0.714 | 0.052 | -13.618 | 0 |

# Pool Results across the 15 imputations

```r
m2_pool <- pool(m2_mods)
summary(m2_pool, conf.int = TRUE, conf.level = 0.95)
```

```
         term     estimate    std.error  statistic        df      p.value
1 (Intercept)  8.284586073 0.0676787686 122.410414  553.4504 0.000000e+00
2     glucose -0.005467924 0.0007996132  -6.838211  475.2087 2.473731e-11
3       obese -0.707891530 0.0526359301 -13.448827 4188.1675 2.132278e-40
         2.5 %       97.5 %
1  8.151647406  8.417524740
2 -0.007039138 -0.003896709
3 -0.811085880 -0.604697181
```

# Model 2 (Complete Cases vs. MI)

```r
tidy(m2_cc, conf.int = T) |> kable(digits = 3) |> kable_styling(font_size =
```

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
|---|---:|---:|---:|---:|---:|---:|
| (Intercept) | 8.259 | 0.066 | 124.78 | 0 | 8.129 | 8.389 |
| glucose | -0.005 | 0.001 | -6.72 | 0 | -0.007 | -0.004 |
| obese | -0.719 | 0.056 | -12.94 | 0 | -0.828 | -0.610 |

```r
summary(m2_pool, conf.int = TRUE, conf.level = 0.95) |>
    select(-df) |> kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | statistic | p.value | 2.5 % | 97.5 % |
|---|---:|---:|---:|---:|---:|---:|
| (Intercept) | 8.285 | 0.068 | 122.410 | 0 | 8.152 | 8.418 |
| glucose | -0.005 | 0.001 | -6.838 | 0 | -0.007 | -0.004 |
| obese | -0.708 | 0.053 | -13.449 | 0 | -0.811 | -0.605 |

# More Details on MI Modeling

```
m2_pool
```

```
Class: mipo    m = 15
         term  m     estimate         ubar            b            t dfcom
1 (Intercept) 15  8.284586073 3.910023e-03 6.284932e-04 4.580416e-03  4235
2     glucose 15 -0.005467924 5.372312e-07 9.576573e-08 6.393813e-07  4235
3       obese 15 -0.707891530 2.758028e-03 1.173120e-05 2.770541e-03  4235
         df        riv      lambda         fmi
1  553.4504 0.17145493 0.146360670 0.149428830
2  475.2087 0.19014180 0.159763988 0.163278086
3 4188.1675 0.00453704 0.004516549 0.004991587
```

Definitions of these terms are in the `mipo` help file.

- `riv` = relative increase in variance attributable to non-response
- `fmi` = fraction of missing information due to non-response

# Model 4 run on each imputed data frame

```r
m4_mods <- with(fram_mice24, lm(inv_sbp ~ glucose +
                              obese + smoker + educ))

summary(m4_mods)
```

```
# A tibble: 105 × 6
   term          estimate std.error statistic  p.value  nobs
   <chr>            <dbl>     <dbl>     <dbl>    <dbl> <int>
 1 (Intercept)    7.99     0.0687      116.   0         4238
 2 glucose       -0.00529  0.000721     -7.34 2.58e-13  4238
 3 obese         -0.625    0.0525      -11.9  3.83e-32  4238
 4 smoker         0.239    0.0349       6.86 7.77e-12  4238
 5 educHS grad    0.209    0.0415       5.05 4.61e- 7  4238
 6 educSome Coll  0.292    0.0504       5.79 7.37e- 9  4238
 7 educColl grad  0.341    0.0579       5.89 4.19e- 9  4238
 8 (Intercept)    8.00     0.0695      115.   0         4238
 9 glucose       -0.00533  0.000727     -7.33 2.76e-13  4238
10 obese         -0.628    0.0526      -11.9  2.50e-32  4238
# … with 95 more rows
```

# Pool Results across the 15 imputations

```r
m4_pool <- pool(m4_mods)

summary(m4_pool, conf.int = TRUE, conf.level = 0.95) |>
    select(-df) |> kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | statistic | p.value | 2.5 % | 97.5 % |
|---|---:|---:|---:|---:|---:|---:|
| (Intercept) | 7.976 | 0.074 | 107.943 | 0 | 7.831 | 8.121 |
| glucose | -0.005 | 0.001 | -6.389 | 0 | -0.007 | -0.003 |
| obese | -0.626 | 0.053 | -11.891 | 0 | -0.730 | -0.523 |
| smoker | 0.240 | 0.035 | 6.858 | 0 | 0.171 | 0.308 |
| educHS grad | 0.198 | 0.042 | 4.701 | 0 | 0.115 | 0.280 |
| educSome Coll | 0.285 | 0.051 | 5.583 | 0 | 0.185 | 0.385 |
| educColl grad | 0.328 | 0.059 | 5.555 | 0 | 0.212 | 0.443 |

# Complete Cases Result (Model 4)

```r
tidy(m4_cc, conf.int = TRUE, conf.level = 0.95) |>
    kable(digits = 3) |> kable_styling(font_size = 28)
```

| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
|---|---:|---:|---:|---:|---:|---:|
| (Intercept) | 7.967 | 0.074 | 107.984 | 0 | 7.822 | 8.111 |
| glucose | -0.005 | 0.001 | -6.382 | 0 | -0.006 | -0.003 |
| obese | -0.650 | 0.057 | -11.472 | 0 | -0.761 | -0.539 |
| smoker | 0.253 | 0.037 | 6.794 | 0 | 0.180 | 0.325 |
| educHS grad | 0.196 | 0.044 | 4.421 | 0 | 0.109 | 0.283 |
| educSome Coll | 0.251 | 0.054 | 4.686 | 0 | 0.146 | 0.357 |
| educColl grad | 0.317 | 0.062 | 5.149 | 0 | 0.196 | 0.438 |

# Additional MI Modeling Details

```
m4_pool
```

```
Class: mipo    m = 15
           term  m     estimate         ubar            b            t dfcom
1   (Intercept) 15  7.976258626 4.773994e-03 6.433669e-04 5.460252e-03  4231
2       glucose 15 -0.005044811 5.269184e-07 9.055590e-08 6.235114e-07  4231
3         obese 15 -0.626379499 2.758532e-03 1.549957e-05 2.775065e-03  4231
4        smoker 15  0.239874783 1.219377e-03 3.749115e-06 1.223376e-03  4231
5   educHS grad 15  0.197535267 1.722712e-03 4.020911e-05 1.765602e-03  4231
6 educSome Coll 15  0.284731274 2.543737e-03 5.364879e-05 2.600963e-03  4231
7 educColl grad 15  0.327760325 3.354429e-03 1.189877e-04 3.481349e-03  4231
         df         riv      lambda         fmi
1  714.9262 0.143749251 0.125682488 0.128118163
2  501.4894 0.183316739 0.154917726 0.158267974
3 4159.4759 0.005993359 0.005957653 0.006435274
4 4201.6596 0.003279589 0.003268868 0.003742976
5 3514.9496 0.024896627 0.024291842 0.024846545
6 3618.4870 0.022496573 0.022001613 0.022541720
```

# Estimate $R^2$ and Adjusted $R^2$

```r
pool.r.squared(m2_mods)
```

```
           est      lo 95      hi 95        fmi
R^2 0.05608137 0.04307754 0.07057275 0.04317449
```

```r
pool.r.squared(m2_mods, adjusted = TRUE)
```

```
               est      lo 95      hi 95        fmi
adj R^2 0.05563553 0.04267941 0.07008282 0.04351396
```

```r
pool.r.squared(m4_mods)
```

```
           est      lo 95      hi 95        fmi
R^2 0.07876698 0.06365358 0.09519128 0.02679134
```

```r
pool.r.squared(m4_mods, adjusted = TRUE)
```

```
               est      lo 95      hi 95        fmi
adj R^2 0.07746049 0.06245732 0.09378374 0.02723597
```

# Tests of Nested Fits after imputation

The models must be nested (same outcome, one set of predictors is a subset of the other) for this to be appropriate.

```r
fit4 <- with(fram_mice24,
             expr = lm(inv_sbp ~ glucose + obese + smoker + educ))
fit2 <- with(fram_mice24,
             expr = lm(inv_sbp ~ glucose + obese))
```

# Comparing Model 4 to Model 2 fits

## We’ll use the Wald test after a linear regression fit.

```
D1(fit4, fit2)
```

```
  test statistic df1      df2 dfcom      p.value       riv
1 ~~ 2  25.43738   4 4049.173  4231 7.634835e-21 0.0230498
```

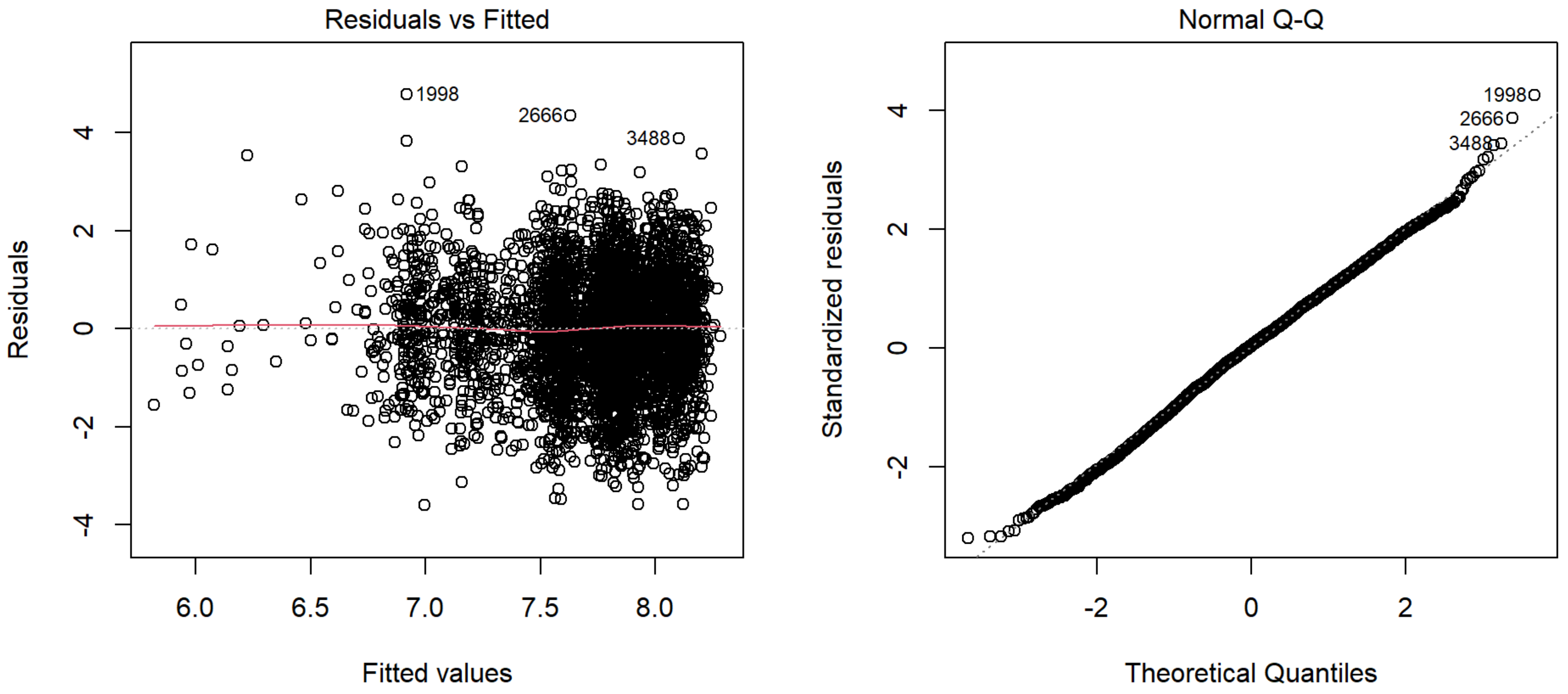
## Could also use a likelihood ratio test.

```
D3(fit4, fit2)
```

```
  test statistic df1      df2 dfcom      p.value        riv
1 ~~ 2  25.20921   4 109075.5  4231 6.674144e-21 0.02152482
```

# Residual Plots for mod4 (6th imputation)

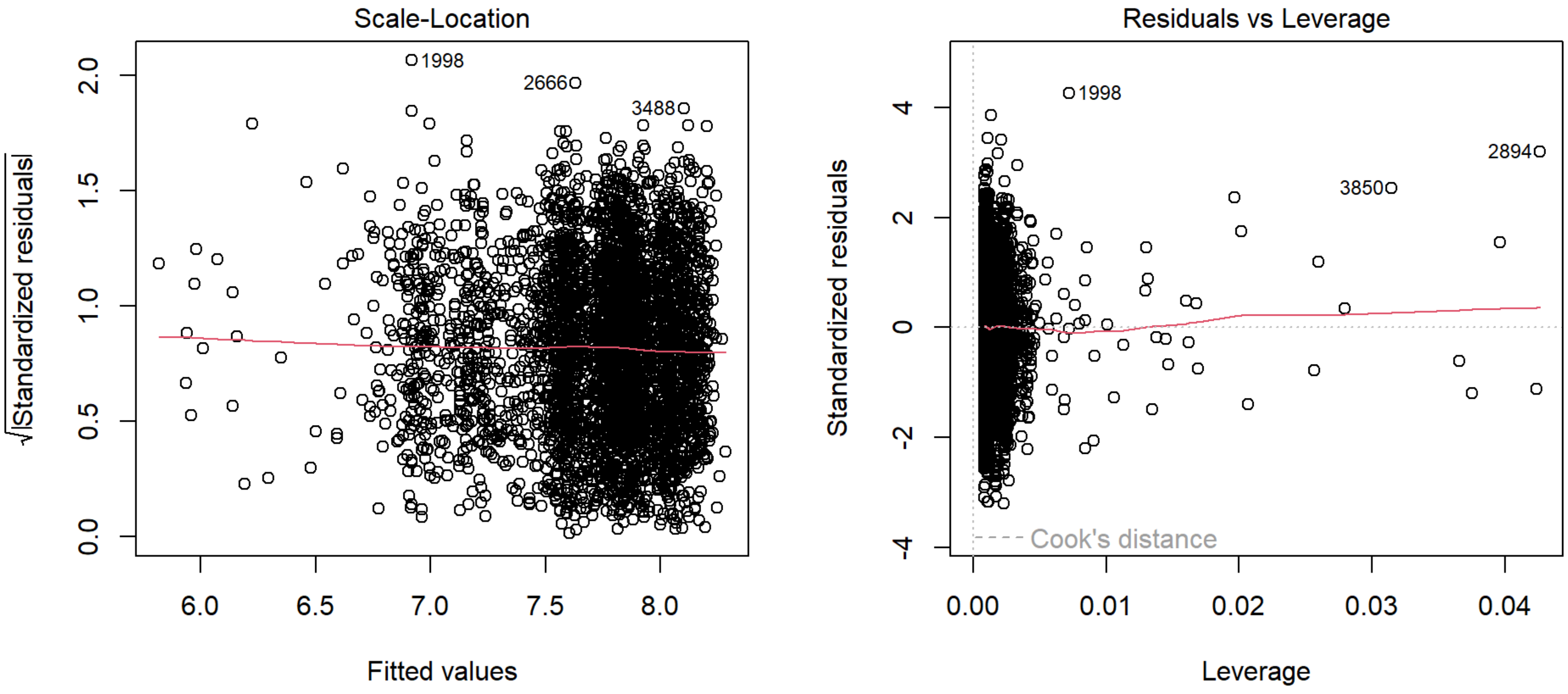
```
par(mfrow = c(1,2))
plot(m4_mods$analyses[[6]], which = c(1:2))
```



# Residual Plots for mod4 (6th imputation)

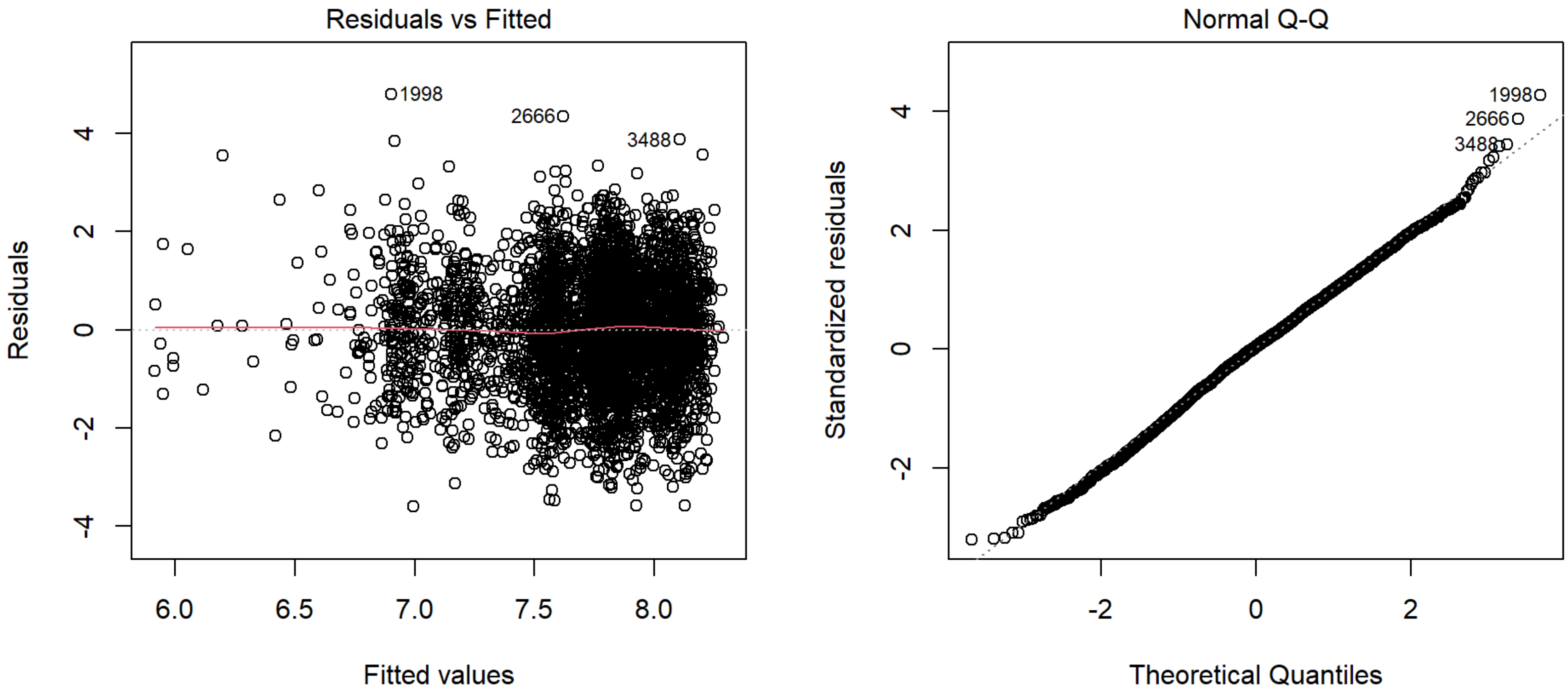
```r
par(mfrow = c(1,2))
plot(m4_mods$analyses[[6]], which = c(3,5))
```

```r
par(mfrow = c(1,1))
```



# Residual Plots for mod4 (1st imputation)

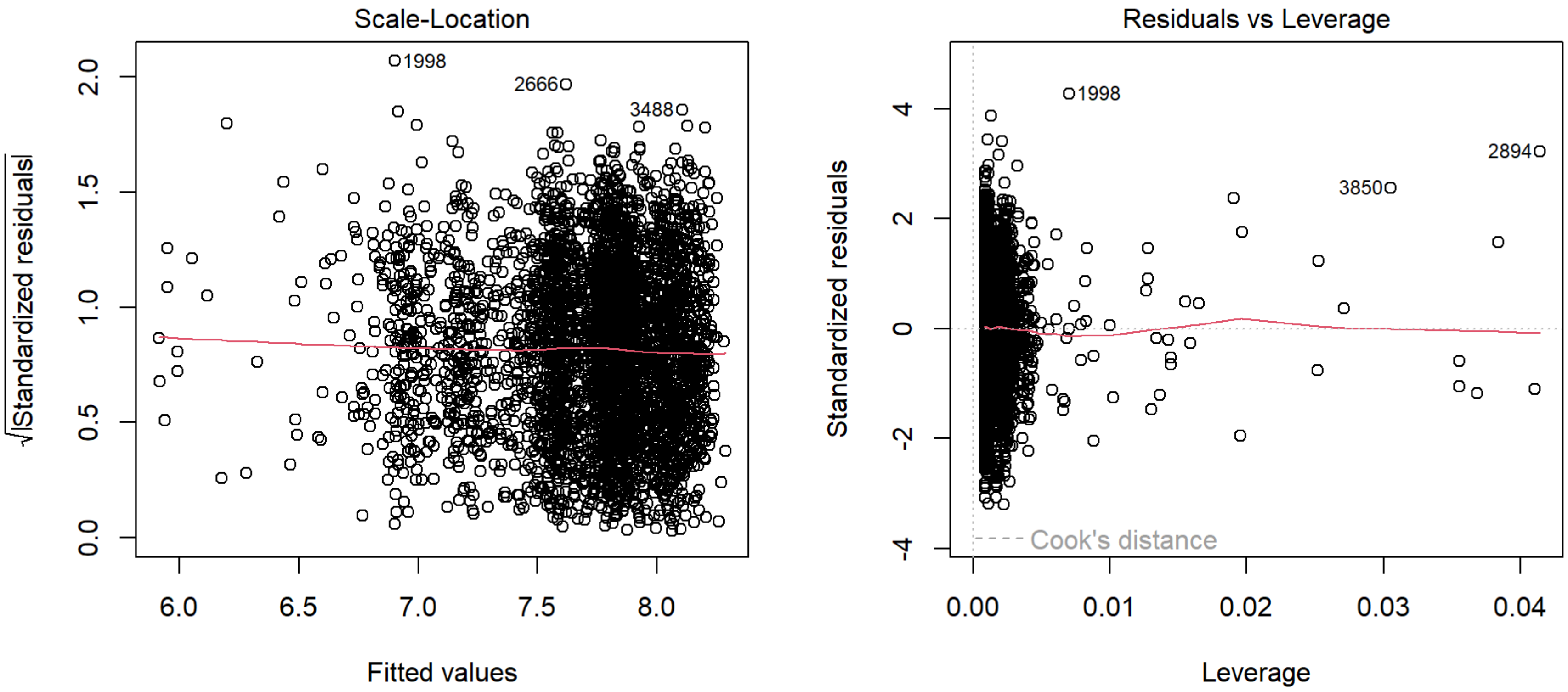
```r
par(mfrow = c(1,2))
plot(m4_mods$analyses[[1]], which = c(1:2))
```



# Residual Plots for mod4 (1st imputation)

```r
par(mfrow = c(1,2))
plot(m4_mods$analyses[[1]], which = c(3,5))
```

```r
par(mfrow = c(1,1))
```



# Guidelines for Reporting

# Guidelines for reporting, I (Sterne et al.)

How should we report on analyses potentially affected by missing data?

- Report the number of missing values for each variable of interest, or the number of cases with complete data for each important component of the analysis. Give reasons for missing values if possible, and indicate how many individuals were excluded because of missing data when reporting the flow of participants through the study. If possible, describe reasons for missing data in terms of other variables (rather than just reporting a universal reason such as treatment failure.)
- Clarify whether there are important differences between individuals with complete and incomplete data, for example, by providing a table comparing the distributions of key exposure and outcome variables in these different groups
- Describe the type of analysis used to account for missing data (eg, multiple imputation), and the assumptions that were made (eg, missing at random)

# Guidelines for reporting, II (Sterne et al.)

How should we report on analyses that involve multiple imputation?

- Provide details of the imputation modeling (software used, key settings, number of imputed datasets, variables included in imputation procedure, etc.)
- If a large fraction of the data is imputed, compare observed and imputed values.
- Where possible, provide results from analyses restricted to complete cases, for comparison with results based on multiple imputation. If there are important differences between the results, suggest explanations.
- It is also desirable to investigate the robustness of key inferences to possible departures from the missing at random assumption, by assuming a range of missing not at random mechanisms in sensitivity analyses.

# Session Information

```
sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```