Prediction model sustained culture conversion

Tu Ha

2025-08-31

```
# load data
cleaned_data_3 <- read_excel("cleaned_data_3.xlsx")

# check missing observations of each variable
cleaned_data_3 %>%
  sapply(function(x)sum(is.na(x)))
```

```
##                 record_id                  MTB_load                 smartt_id
##                         0                         2                         0
##                    bl_age                pretx_sex               ptretx_bmi
##                         0                         0                         0
##       ses_education_level living_alone_37eb74_v2_v2       smoker_5c21df_v2_v2
##                        31                         6                         6
##      alcohol_83d0af_v2_v2                 tretx_dm          prettx_prevtbtx
##                         8                         0                         0
##   late_culture_conversion          time_culture_pos                 pretx_hiv
##                         0                         0                         0
##        hiv_status_control                        hb        qol_usual_activity
##                         4                         4                         1
##                 edu_level         ses_income_before                  INH_fill
##                        11                        32                         3
##                  FQs_fill        resistance_pattern
##                         4                         4
```

```
# code all categories as 1,2,3...
cleaned_data_4 <- cleaned_data_3 %>%
  mutate(MTB_load = case_when(
    MTB_load == "very low" ~ 1,
    MTB_load == "low" ~ 2,
    MTB_load == "medium" ~ 3,
    MTB_load == "high" ~ 4,
    TRUE ~ NA_real_
  ),
  resistance_pattern = case_when(
    resistance_pattern == "monoDR" ~ 1,
    resistance_pattern == "MDR" ~ 2,
    resistance_pattern == "(pre)XDR" ~ 3,
    TRUE ~ NA_real_
  ))
```

1. Explore data

```
table(cleaned_data_4$MTB_load, cleaned_data_4$late_culture_conversion, useNA = "always")
```

```
##
##         0  1 <NA>
##   1    10 11    0
##   2    20  8    0
##   3    16 23    0
##   4    26 53    0
##   <NA>  1  1    0
```

```
round(prop.table(table(cleaned_data_3$MTB_load, cleaned_data_3$late_culture_conversion, useNA = "always
```

```
##
##                0     1  <NA>
##   high     0.154 0.314 0.000
##   low      0.118 0.047 0.000
##   medium   0.095 0.136 0.000
##   very low 0.059 0.065 0.000
##   <NA>     0.006 0.006 0.000
```

```
table(cleaned_data_4$alcohol_83d0af_v2_v2, cleaned_data_4$late_culture_conversion, useNA = "always")
```

```
##
##         0  1 <NA>
##   1    26 40    0
##   2    22 24    0
##   3    16 19    0
##   4     6  8    0
##   <NA>  3  5    0
```

```
round(prop.table(table(cleaned_data_3$MTB_load, cleaned_data_3$late_culture_conversion, useNA = "always
```

```
##
##                0     1  <NA>
##   high     0.154 0.314 0.000
##   low      0.118 0.047 0.000
##   medium   0.095 0.136 0.000
##   very low 0.059 0.065 0.000
##   <NA>     0.006 0.006 0.000
```

1 None, 2 Light (once a month), 3 Moderate (once a week), 4 Heavy (daily)

```
table(cleaned_data_3$ses_education_level, cleaned_data_3$late_culture_conversion, useNA = "always")
```

```
##
##         0  1 <NA>
##   1     7 11    0
##   2    38 49    0
##   3    16 17    0
##   <NA> 12 19    0
```

```
round(prop.table(table(cleaned_data_3$ses_education_level, cleaned_data_3$late_culture_conversion, useNA
```

```
##
##            0     1  <NA>
##   1    0.041 0.065 0.000
##   2    0.225 0.290 0.000
##   3    0.095 0.101 0.000
##   <NA> 0.071 0.112 0.000
```

1 No secondary education, 2 Some secondary education, 3 Matric or higher

```
table(cleaned_data_3$edu_level, cleaned_data_3$late_culture_conversion, useNA = "always")
```

```
##
##         0  1 <NA>
##   1     2  2    0
##   2    19 15    0
##   3    47 69    0
##   4     2  2    0
##   <NA>  3  8    0
```

```
round(prop.table(table(cleaned_data_3$edu_level, cleaned_data_3$late_culture_conversion, useNA = "always
```

```
##
##            0     1  <NA>
##   1    0.012 0.012 0.000
##   2    0.112 0.089 0.000
##   3    0.278 0.408 0.000
##   4    0.012 0.012 0.000
##   <NA> 0.018 0.047 0.000
```

1: no schooling, 2: primary, 3: secondary, 4: tertiary

```
table(cleaned_data_3$hiv_status_control, cleaned_data_3$late_culture_conversion, useNA = "always")
```

```
##
##         0  1 <NA>
##   0    28 32    0
##   1    12 32    0
##   2    16 15    0
##   3    15 15    0
##   <NA>  2  2    0
```

```
round(prop.table(table(cleaned_data_3$hiv_status_control, cleaned_data_3$late_culture_conversion, useNA
```

```
##
##            0     1  <NA>
##   0    0.166 0.189 0.000
##   1    0.071 0.189 0.000
##   2    0.095 0.089 0.000
##   3    0.089 0.089 0.000
##   <NA> 0.012 0.012 0.000
```

0 HIV negative, 1 HIV positive, on ART and viral load controlled (Viral load < = 1000), 2 HIV positive, on ART with no viral load control (Viral load >1000), 3 HIV positive, but not on ART

```
table(cleaned_data_3$resistance_pattern, cleaned_data_3$late_culture_conversion, useNA = "always")
```

```
##
##              0  1 <NA>
##   (pre)XDR   6 13    0
##   MDR       23 33    0
##   monoDR    41 49    0
##   <NA>       3  1    0
```

```
round(prop.table(table(cleaned_data_3$resistance_pattern, cleaned_data_3$late_culture_conversion, useNA
```

```
##
##                 0     1  <NA>
##   (pre)XDR  0.036 0.077 0.000
##   MDR       0.136 0.195 0.000
##   monoDR    0.243 0.290 0.000
##   <NA>      0.018 0.006 0.000
```

```
table(cleaned_data_3$qol_usual_activity, cleaned_data_3$late_culture_conversion, useNA = "always")
```

```
##
##          0  1 <NA>
##   1     52 48    0
##   2     12 29    0
##   3      6 11    0
##   4      3  5    0
##   5      0  2    0
##   <NA>   0  1    0
```

```
round(prop.table(table(cleaned_data_3$qol_usual_activity, cleaned_data_3$late_culture_conversion, useNA
```

```
##
##             0     1  <NA>
##   1     0.308 0.284 0.000
##   2     0.071 0.172 0.000
##   3     0.036 0.065 0.000
##   4     0.018 0.030 0.000
##   5     0.000 0.012 0.000
##   <NA>  0.000 0.006 0.000
```

1: I have no problems doing my usual activities, 2: I have slight problems doing my usual activities, 3: I have moderate problems doing my usual activities, 4: I have severe problems doing my usual activities, 5: I am unable to do my usual activities first dataset

```
# create data for imputation
data_1_pre_impute <- cleaned_data_4 %>%
  select(-record_id, -smartt_id, -edu_level, -hiv_status_control, -INH_fill, -FQs_fill )

# percentage of rows containing at least 1 NA
1- nrow(na.omit(data_1_pre_impute))/ nrow(data_1_pre_impute)
```

```
## [1] 0.2899408
```

```
# => 29% of rows with any NAs
```

```
data_1_pre_impute %>% sapply(function(x)sum(is.na(x)))
```

```
##                   MTB_load                     bl_age                  pretx_sex
##                          2                          0                          0
##                ptretx_bmi       ses_education_level living_alone_37eb74_v2_v2
##                          0                         31                          6
##       smoker_5c21df_v2_v2      alcohol_83d0af_v2_v2                   tretx_dm
##                          6                          8                          0
##           prettx_prevtbtx   late_culture_conversion          time_culture_pos
##                          0                          0                          0
##                 pretx_hiv                         hb        qol_usual_activity
##                          0                          4                          1
##         ses_income_before        resistance_pattern
##                         32                          4
```

```
#str(data_1_pre_impute)
#summary(data_1_pre_impute)
```

Change class for variables before imputation

```
data_1_pre_impute<- data_1_pre_impute %>%
  mutate(
    MTB_load = as.factor(MTB_load),
    pretx_sex = as.factor(pretx_sex),
    ses_education_level = as.factor(ses_education_level),
    living_alone_37eb74_v2_v2 = as.factor(living_alone_37eb74_v2_v2),
    smoker_5c21df_v2_v2 = as.factor(smoker_5c21df_v2_v2),
    alcohol_83d0af_v2_v2 = as.factor(alcohol_83d0af_v2_v2),
    tretx_dm = as.factor(tretx_dm),
    prettx_prevtbtx = as.factor(prettx_prevtbtx),
    late_culture_conversion = as.factor(late_culture_conversion),
    pretx_hiv = as.factor(pretx_hiv),
    resistance_pattern = as.factor(resistance_pattern),
    qol_usual_activity = as.factor(qol_usual_activity)
  )

# Check the structure to ensure the changes
str(data_1_pre_impute)
```

```
## tibble [169 x 17] (S3: tbl_df/tbl/data.frame)
##  $ MTB_load                 : Factor w/ 4 levels "1","2","3","4": 1 4 1 4 4 3 3 4 4 3 ...
##  $ bl_age                   : num [1:169] 61.4 32.6 59.1 33.1 22 38.8 30.4 44 42.6 68.8 ...
##  $ pretx_sex                : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 1 2 1 ...
##  $ ptretx_bmi               : num [1:169] 18.8 18.2 22.8 16.1 16.2 ...
##  $ ses_education_level      : Factor w/ 3 levels "1","2","3": 1 2 3 NA 2 2 2 NA 2 2 ...
##  $ living_alone_37eb74_v2_v2: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ smoker_5c21df_v2_v2      : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 1 2 1 ...
##  $ alcohol_83d0af_v2_v2     : Factor w/ 4 levels "1","2","3","4": 2 1 1 2 1 2 1 1 1 1 ...
```

```
##  $ tretx_dm                : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 1 2 ...
##  $ prettx_prevtbtx         : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 2 1 1 1 ...
##  $ late_culture_conversion : Factor w/ 2 levels "0","1": 2 2 1 1 2 1 2 1 1 2 ...
##  $ time_culture_pos        : num [1:169] 11 6 15 18 13 21 15 13 21 20 ...
##  $ pretx_hiv               : Factor w/ 2 levels "0","1": 2 1 1 2 1 1 2 2 1 1 ...
##  $ hb                      : num [1:169] 9.4 13.5 12.2 7 12 9.2 13.3 11.3 12 10.4 ...
##  $ qol_usual_activity      : Factor w/ 5 levels "1","2","3","4",..: 1 2 1 4 1 2 2 2 1 4 ...
##  $ ses_income_before       : num [1:169] 5000 3500 6000 NA 1200 0 600 NA 1800 1850 ...
##  $ resistance_pattern      : Factor w/ 3 levels "1","2","3": 1 1 2 1 1 2 1 2 1 1 ...
```

summarise data

```
#percentage are calculated after excluding NAs
tab1 <- CreateTableOne(data = data_1_pre_impute)
tab1_df <- print(tab1, showAllLevels = T)
```

```
##
##                                level Overall
##   n                                      169
##   MTB_load (%)                 1          21 (12.6)
##                                2          28 (16.8)
##                                3          39 (23.4)
##                                4          79 (47.3)
##   bl_age (mean (SD))                   41.55 (12.77)
##   pretx_sex (%)                1          46 (27.2)
##                                2         123 (72.8)
##   ptretx_bmi (mean (SD))               18.70 (4.10)
##   ses_education_level (%)      1          18 (13.0)
##                                2          87 (63.0)
##                                3          33 (23.9)
##   living_alone_37eb74_v2_v2 (%) 0        144 (88.3)
##                                1          19 (11.7)
##   smoker_5c21df_v2_v2 (%)      0          78 (47.9)
##                                1          85 (52.1)
##   alcohol_83d0af_v2_v2 (%)     1          66 (41.0)
##                                2          46 (28.6)
##                                3          35 (21.7)
##                                4          14 ( 8.7)
##   tretx_dm (%)                 0         157 (92.9)
##                                1          12 ( 7.1)
##   prettx_prevtbtx (%)          0          97 (57.4)
##                                1          72 (42.6)
##   late_culture_conversion (%)  0          73 (43.2)
##                                1          96 (56.8)
##   time_culture_pos (mean (SD))         16.69 (11.54)
##   pretx_hiv (%)                0          60 (35.5)
##                                1         109 (64.5)
##   hb (mean (SD))                       11.22 (2.38)
##   qol_usual_activity (%)       1         100 (59.5)
##                                2          41 (24.4)
##                                3          17 (10.1)
##                                4           8 ( 4.8)
##                                5           2 ( 1.2)
```

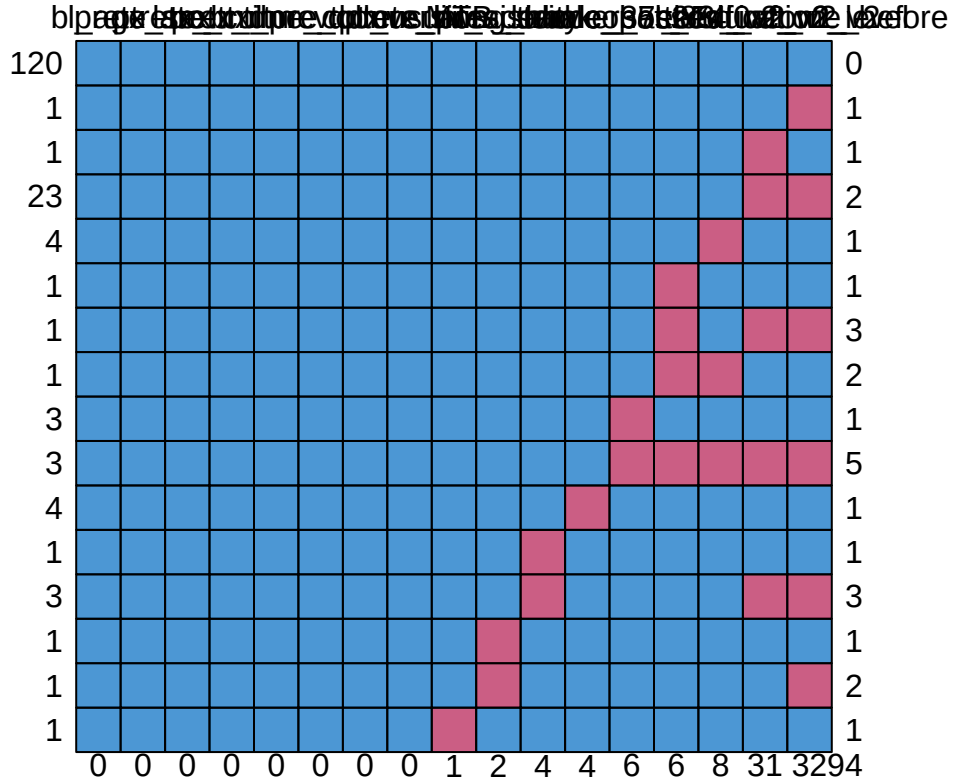
```
##   ses_income_before (mean (SD))       3835.73 (5802.57)
##   resistance_pattern (%)        1          90 (54.5)
##                                 2          56 (33.9)
##                                 3          19 (11.5)
```

```
#summarize, with NAs included
summary(tab1)
```

```
##
##      ### Summary of continuous variables ###
##
## strata: Overall
##                      n miss p.miss mean   sd median  p25  p75 min   max skew
## bl_age             169    0      0   42   13     40   32   49  18    75  0.4
## ptretx_bmi         169    0      0   19    4     18   16   20  12    38  1.7
## time_culture_pos   169    0      0   17   12     13    9   21   2    42  1.2
## hb                 169    4      2   11    2     11   10   13   4    18 -0.1
## ses_income_before  169   32     19 3836 5803   2600 1350 4500   0 60000  7.1
##                     kurt
## bl_age             -0.37
## ptretx_bmi          4.51
## time_culture_pos    0.33
## hb                  0.03
## ses_income_before  65.43
##
## ============================================================================================
##
##      ### Summary of categorical variables ###
##
## strata: Overall
##                          var   n miss p.miss level freq percent cum.percent
##                    MTB_load 169    2    1.2     1   21    12.6        12.6
##                                                2   28    16.8        29.3
##                                                3   39    23.4        52.7
##                                                4   79    47.3       100.0
##
##                   pretx_sex 169    0    0.0     1   46    27.2        27.2
##                                                2  123    72.8       100.0
##
##         ses_education_level 169   31   18.3     1   18    13.0        13.0
##                                                2   87    63.0        76.1
##                                                3   33    23.9       100.0
##
##  living_alone_37eb74_v2_v2 169    6    3.6     0  144    88.3        88.3
##                                                1   19    11.7       100.0
##
##         smoker_5c21df_v2_v2 169    6    3.6     0   78    47.9        47.9
##                                                1   85    52.1       100.0
##
##       alcohol_83d0af_v2_v2 169    8    4.7     1   66    41.0        41.0
##                                                2   46    28.6        69.6
##                                                3   35    21.7        91.3
##                                                4   14     8.7       100.0
##
```

```
##                   tretx_dm 169    0    0.0     0  157    92.9        92.9
##                                                1   12     7.1       100.0
##
##            prettx_prevtbtx 169    0    0.0     0   97    57.4        57.4
##                                                1   72    42.6       100.0
##
##    late_culture_conversion 169    0    0.0     0   73    43.2        43.2
##                                                1   96    56.8       100.0
##
##                  pretx_hiv 169    0    0.0     0   60    35.5        35.5
##                                                1  109    64.5       100.0
##
##         qol_usual_activity 169    1    0.6     1  100    59.5        59.5
##                                                2   41    24.4        83.9
##                                                3   17    10.1        94.0
##                                                4    8     4.8        98.8
##                                                5    2     1.2       100.0
##
##         resistance_pattern 169    4    2.4     1   90    54.5        54.5
##                                                2   56    33.9        88.5
##                                                3   19    11.5       100.0
##
```

```
md.pattern(data_1_pre_impute, plot = T)
```




```
##     bl_age pretx_sex ptretx_bmi tretx_dm prettx_prevtbtx
```

```
## 120      1         1          1        1               1
## 1        1         1          1        1               1
## 1        1         1          1        1               1
## 23       1         1          1        1               1
## 4        1         1          1        1               1
## 1        1         1          1        1               1
## 1        1         1          1        1               1
## 1        1         1          1        1               1
## 3        1         1          1        1               1
## 3        1         1          1        1               1
## 4        1         1          1        1               1
## 1        1         1          1        1               1
## 3        1         1          1        1               1
## 1        1         1          1        1               1
## 1        1         1          1        1               1
## 1        1         1          1        1               1
##          0         0          0        0               0
##     late_culture_conversion time_culture_pos pretx_hiv qol_usual_activity
## 120                       1               1         1                   1
## 1                         1               1         1                   1
## 1                         1               1         1                   1
## 23                        1               1         1                   1
## 4                         1               1         1                   1
## 1                         1               1         1                   1
## 1                         1               1         1                   1
## 1                         1               1         1                   1
## 3                         1               1         1                   1
## 3                         1               1         1                   1
## 4                         1               1         1                   1
## 1                         1               1         1                   1
## 3                         1               1         1                   1
## 1                         1               1         1                   1
## 1                         1               1         1                   1
## 1                         1               1         1                   0
##                           0               0         0                   1
##     MTB_load hb resistance_pattern living_alone_37eb74_v2_v2
## 120        1  1                  1                         1
## 1          1  1                  1                         1
## 1          1  1                  1                         1
## 23         1  1                  1                         1
## 4          1  1                  1                         1
## 1          1  1                  1                         1
## 1          1  1                  1                         1
## 1          1  1                  1                         1
## 3          1  1                  1                         0
## 3          1  1                  1                         0
## 4          1  1                  0                         1
## 1          1  0                  1                         1
## 3          1  0                  1                         1
## 1          0  1                  1                         1
## 1          0  1                  1                         1
## 1          1  1                  1                         1
##            2  4                  4                         6
##     smoker_5c21df_v2_v2 alcohol_83d0af_v2_v2 ses_education_level
```

```
## 120                   1                   1                   1
## 1                     1                   1                   1
## 1                     1                   1                   0
## 23                    1                   1                   0
## 4                     1                   0                   1
## 1                     0                   1                   1
## 1                     0                   1                   0
## 1                     0                   0                   1
## 3                     1                   1                   1
## 3                     0                   0                   0
## 4                     1                   1                   1
## 1                     1                   1                   1
## 3                     1                   1                   0
## 1                     1                   1                   1
## 1                     1                   1                   1
## 1                     1                   1                   1
##                       6                   8                  31
##     ses_income_before
## 120                 1  0
## 1                   0  1
## 1                   1  1
## 23                  0  2
## 4                   1  1
## 1                   1  1
## 1                   0  3
## 1                   1  2
## 3                   1  1
## 3                   0  5
## 4                   1  1
## 1                   1  1
## 3                   0  3
## 1                   1  1
## 1                   0  2
## 1                   1  1
##                    32 94
```

test for MCAR

```
mcar_test(data_1_pre_impute)
```

```
## # A tibble: 1 x 4
##   statistic    df p.value missing.patterns
##       <dbl> <dbl>   <dbl>            <int>
## 1      256.   229   0.107               16
```

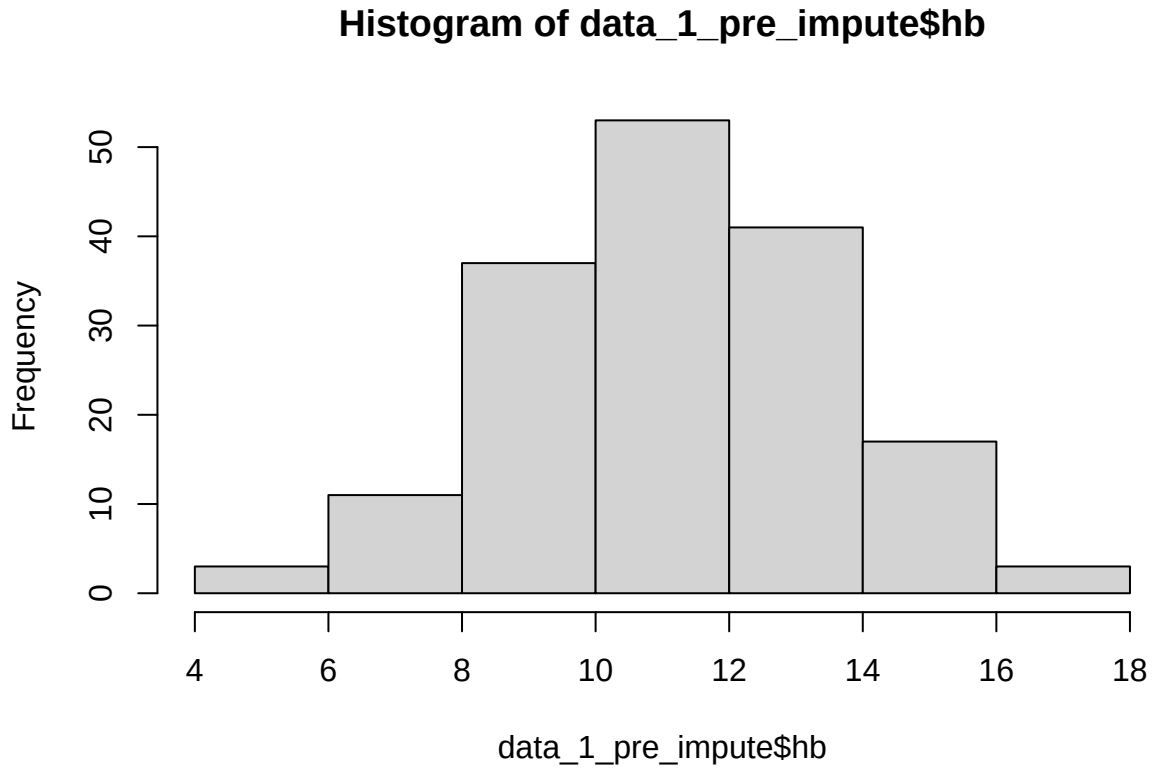
p-value =0.107 > 0.05, fail to reject H0 -> no evidence against MCAR

2. Multiple imputation

```
ini_d1 <- mice(data_1_pre_impute, m=30, maxit=0, seed = 111, print=FALSE)
ini_d1$method
```

```
##                  MTB_load                    bl_age                 pretx_sex
##                "polyreg"                        ""                        ""
##                ptretx_bmi       ses_education_level living_alone_37eb74_v2_v2
##                        ""                "polyreg"                 "logreg"
##       smoker_5c21df_v2_v2      alcohol_83d0af_v2_v2                 tretx_dm
##                 "logreg"                "polyreg"                        ""
##           prettx_prevtbtx   late_culture_conversion          time_culture_pos
##                        ""                        ""                        ""
##                pretx_hiv                        hb        qol_usual_activity
##                        ""                    "pmm"                 "polyreg"
##         ses_income_before        resistance_pattern
##                     "pmm"                 "polyreg"
```

```
hist(data_1_pre_impute$hb)
```

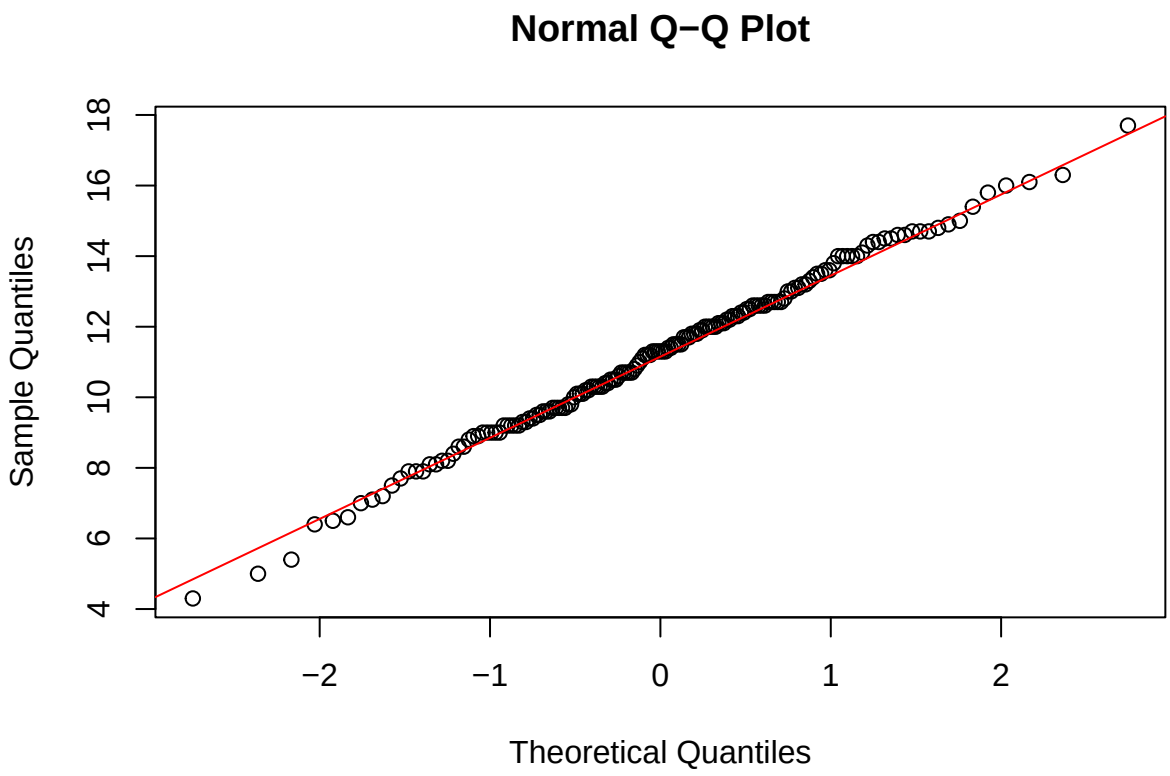



```
shapiro.test(data_1_pre_impute$hb)
```

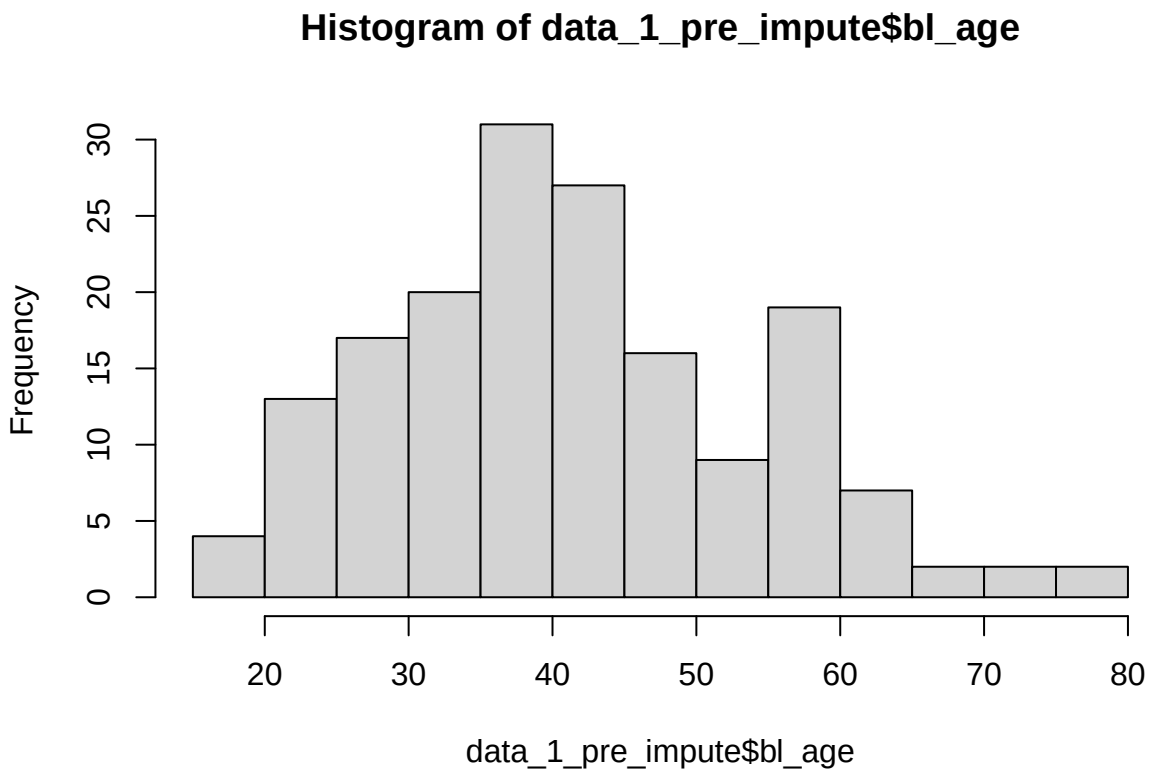
```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  data_1_pre_impute$hb
## W = 0.99617, p-value = 0.9513
```

```r
# => p-value >0.05 ->Hb normally distributed

qqnorm(data_1_pre_impute$hb)
qqline(data_1_pre_impute$hb, col = "red")
```



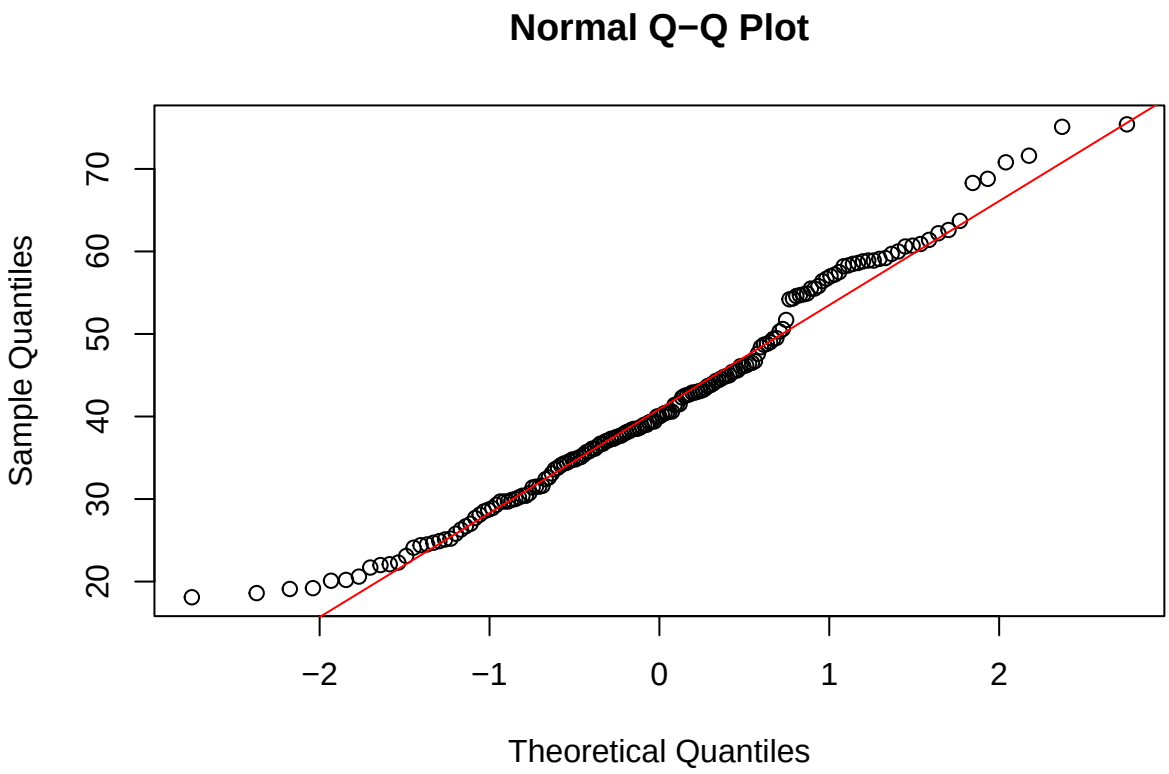

```r
#=> for hb, use norm in stead of pmm for more efficiency

#summary(data_1_pre_impute)

hist(data_1_pre_impute$bl_age)
```




```
shapiro.test(data_1_pre_impute$bl_age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_1_pre_impute$bl_age
## W = 0.97727, p-value = 0.00709
```

```
qqnorm(data_1_pre_impute$bl_age)
qqline(data_1_pre_impute$bl_age, col = "red")
```



For sparse categorical data, it may be better to use method pmm instead of logreg, polr or polyreg. Method logreg.boot is a version of logreg that uses the bootstrap to emulate sampling variance. [qol_usual_activity, alcohol_83d0af_v2_v2, resistance_pattern, ses_education_level]

```
meth <- ini_d1$method
meth["qol_usual_activity"] <- "pmm"
meth["alcohol_83d0af_v2_v2"] <- "pmm"
meth["resistance_pattern"] <- "pmm"
meth["ses_education_level"] <- "pmm"
meth["living_alone_37eb74_v2_v2"] <- "pmm"
meth
```

```
##                MTB_load                  bl_age                pretx_sex
##              "polyreg"                      ""                      ""
##             ptretx_bmi     ses_education_level living_alone_37eb74_v2_v2
##                      ""                   "pmm"                   "pmm"
##     smoker_5c21df_v2_v2    alcohol_83d0af_v2_v2                tretx_dm
##              "logreg"                   "pmm"                      ""
##         prettx_prevtbtx late_culture_conversion         time_culture_pos
##                      ""                      ""                      ""
##              pretx_hiv                      hb       qol_usual_activity
##                      ""                   "pmm"                   "pmm"
##       ses_income_before      resistance_pattern
##                   "pmm"                   "pmm"
```

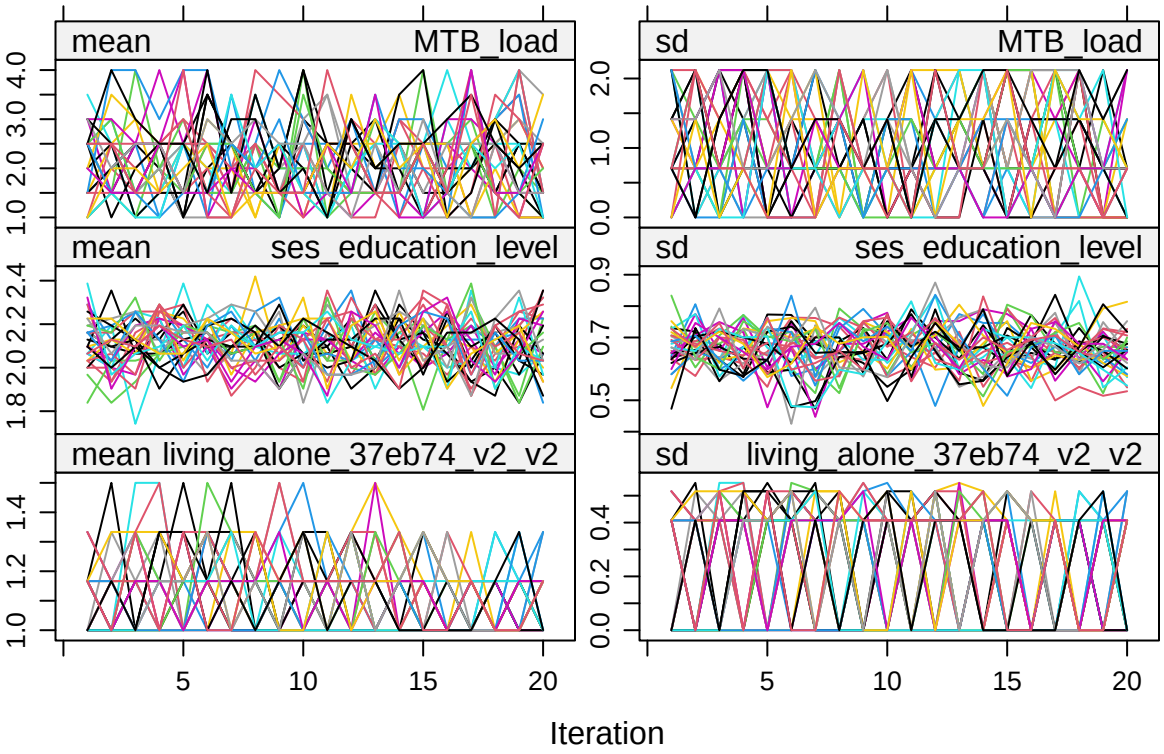
```r
pred_d1 <- ini_d1$predictorMatrix
# Set time_culture_pos as an auxiliary variable for MTB_load
pred_d1["MTB_load", "time_culture_pos"] <- 1
# time_culture_pos is not used for imputing other variables
other_vars <- setdiff(rownames(pred_d1), "MTB_load")
pred_d1[other_vars, "time_culture_pos"] <- 0
```

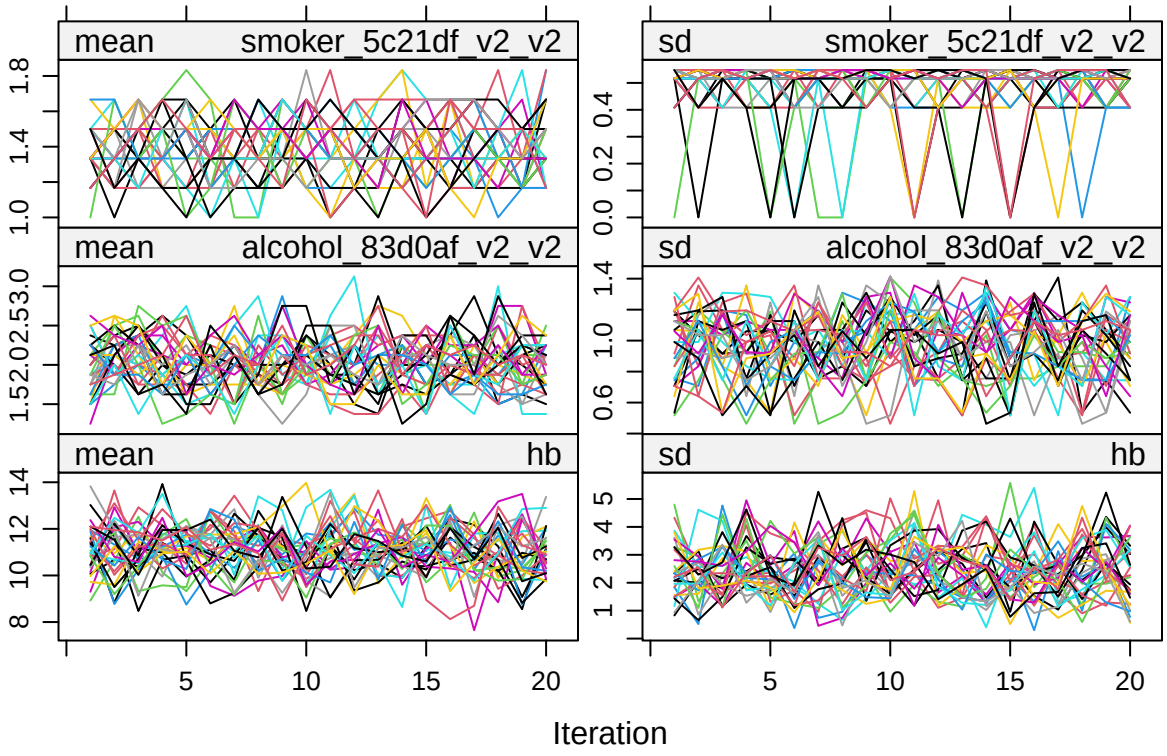
```r
# as number of missing data ~30% -> use m= 30 (number of imputations)
imp1_d2 <- mice(data_1_pre_impute,
             pred=pred_d1,method = meth, m=30, maxit = 20, print =FALSE, seed= 111)

# check types of imputation for each variable
imp1_d2$method
```

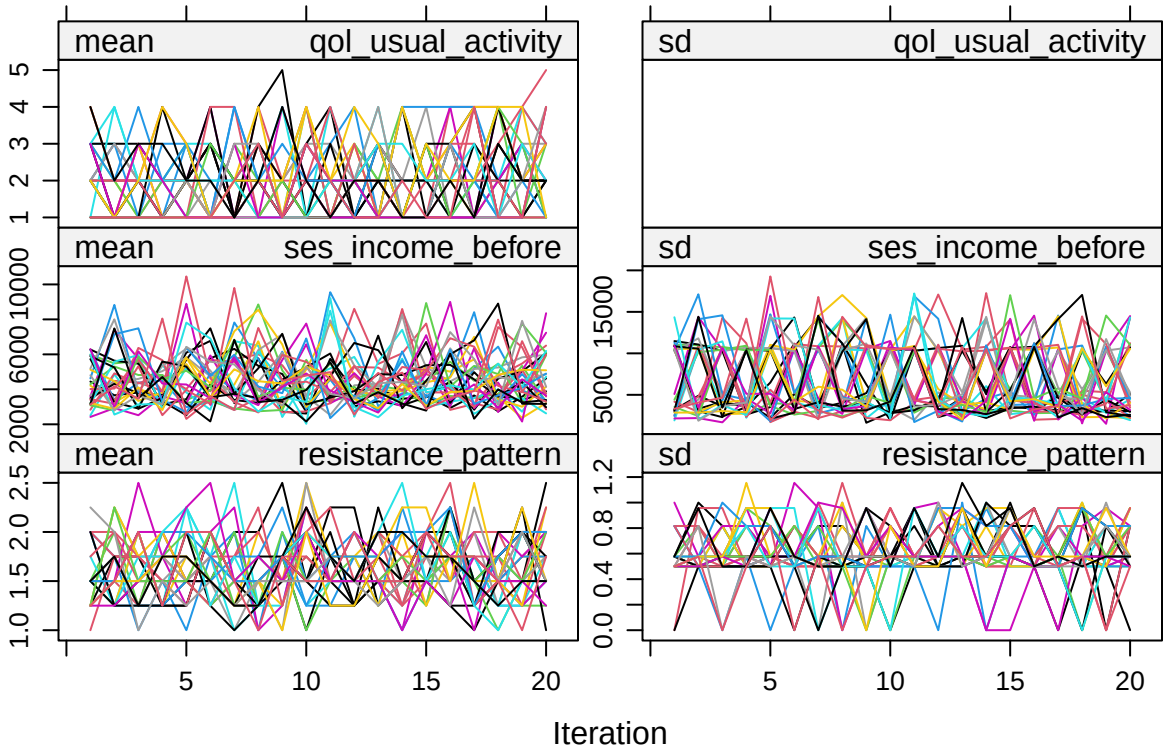
```
##                  MTB_load                    bl_age                  pretx_sex
##                "polyreg"                        ""                        ""
##               ptretx_bmi       ses_education_level living_alone_37eb74_v2_v2
##                        ""                     "pmm"                     "pmm"
##       smoker_5c21df_v2_v2      alcohol_83d0af_v2_v2                  tretx_dm
##                  "logreg"                     "pmm"                        ""
##           prettx_prevtbtx   late_culture_conversion          time_culture_pos
##                        ""                        ""                        ""
##                 pretx_hiv                        hb        qol_usual_activity
##                        ""                     "pmm"                     "pmm"
##         ses_income_before        resistance_pattern
##                     "pmm"                     "pmm"
```

Check convergence

```r
plot(imp1_d2)
```


mean MTB_load
sd MTB_load
mean ses_education_level
sd ses_education_level
mean living_alone_37eb74_v2_v2
sd living_alone_37eb74_v2_v2
1.0 2.0 3.0 4.0
0.0 1.0 2.0
1.8 2.0 2.2 2.4
0.5 0.7 0.9
1.0 1.2 1.4
0.0 0.2 0.4
5 10 15 20
Iteration


mean
smoker_5c21df_v2_v2
mean
alcohol_83d0af_v2_v2
mean
hb
sd
smoker_5c21df_v2_v2
sd
alcohol_83d0af_v2_v2
sd
hb
Iteration

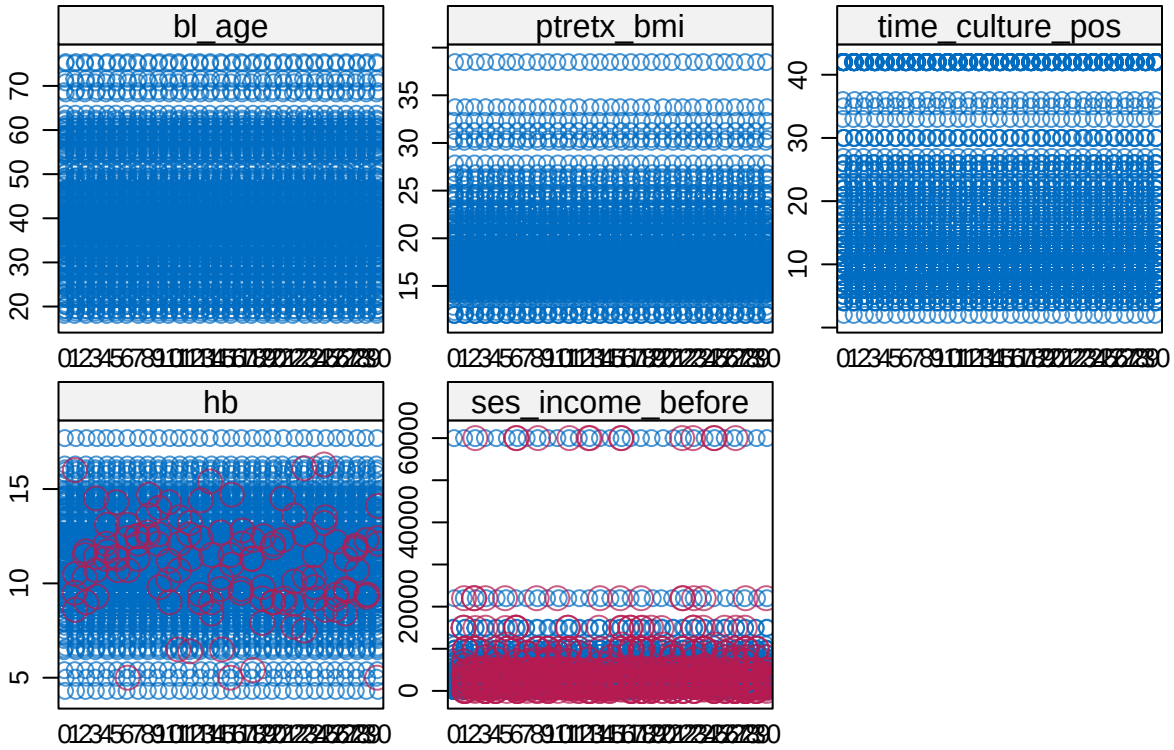


3. Diagnosis MI

In imputation it is often more informative to focus on distributional discrepancy, the difference between the observed and imputed data

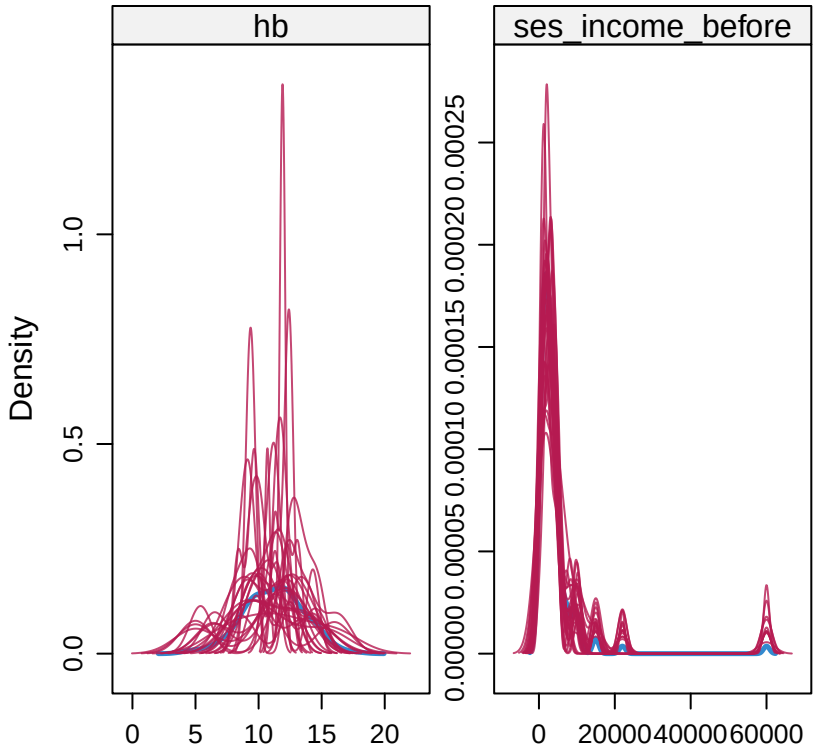
The idea is that good imputations have a distribution similar to the observed data. In other words, the imputations could have been real values had they been observed. Except under MCAR, the distributions do not need to be identical, since strong MAR mechanisms may induce systematic differences between the two distributions. However, any dramatic differences between the imputed and observed data should certainly alert us to the possibility that something is wrong.

```
stripplot(imp1_d2, cex = c(1, 1.5))
```



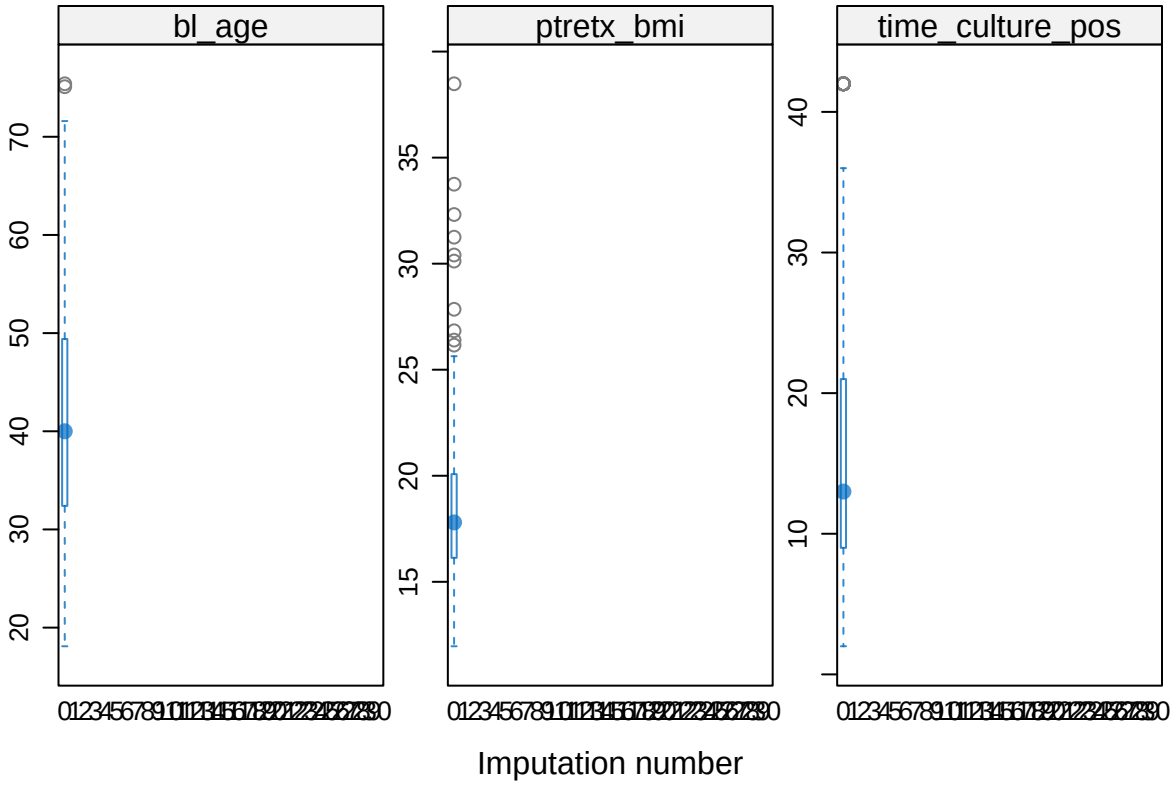
Kernel density estimates for the marginal distributions of the observed data (blue) and the m=30 densities per variable calculated from the imputed data (thin red lines).

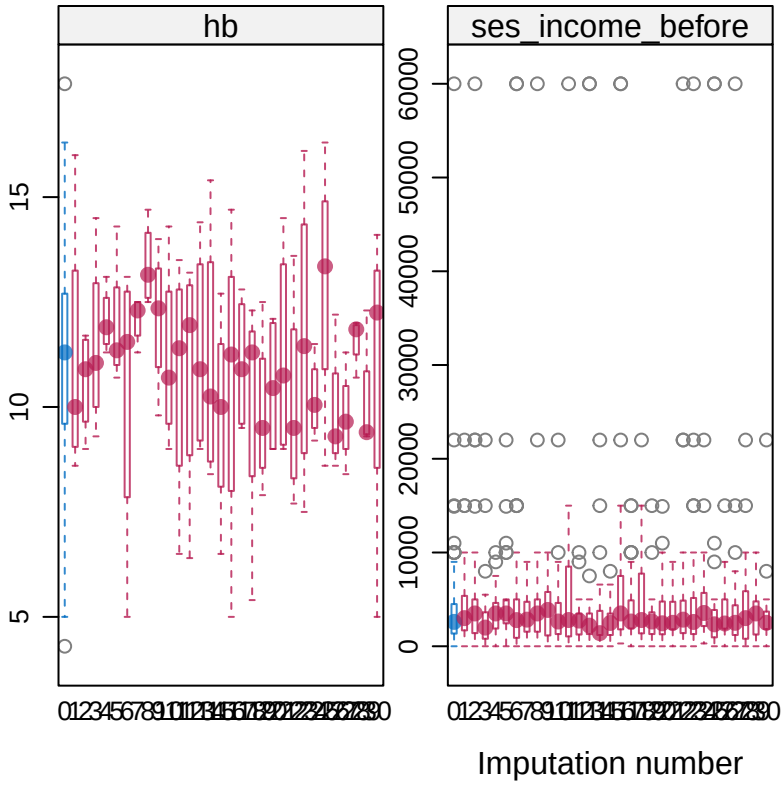
```
densityplot(imp1_d2, layout=c(3,1))
```



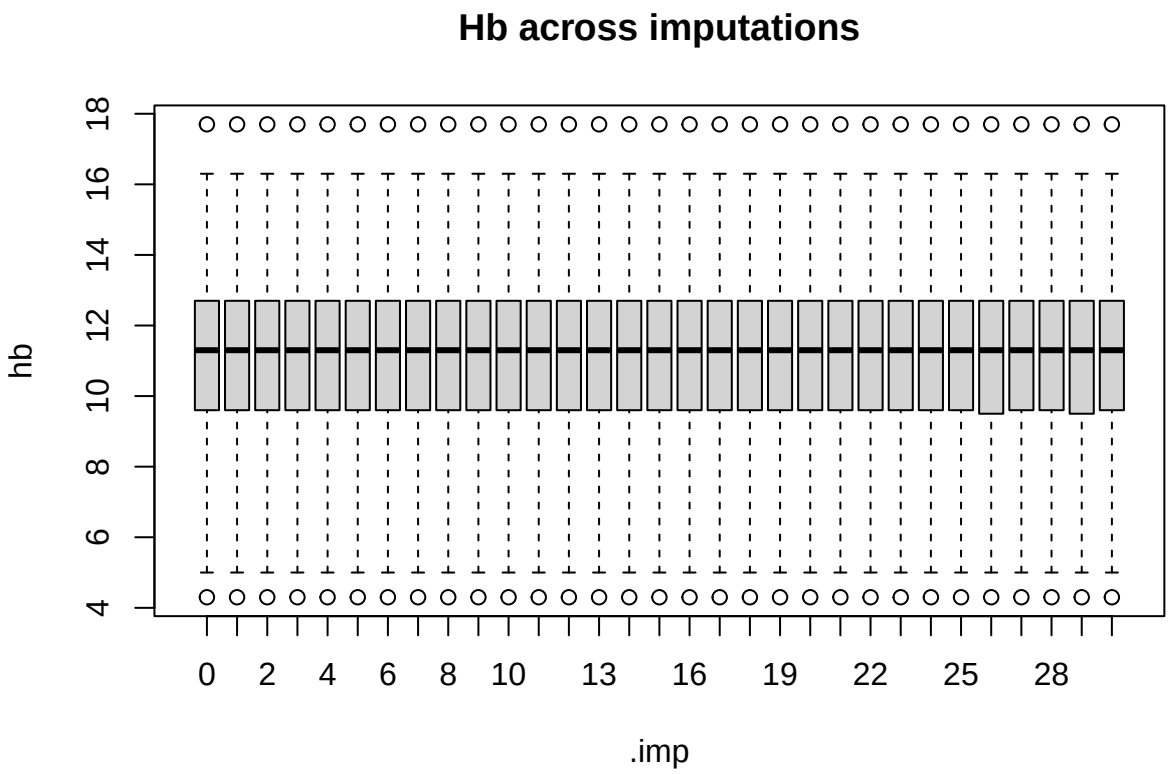
Interpretation is more difficult if there are discrepancies. Such discrepancies may be caused by a bad imputation model, by a missing data mechanism that is not MCAR or by a combination of both

```
bwplot(imp1_d2, layout = c(3, 1))
```

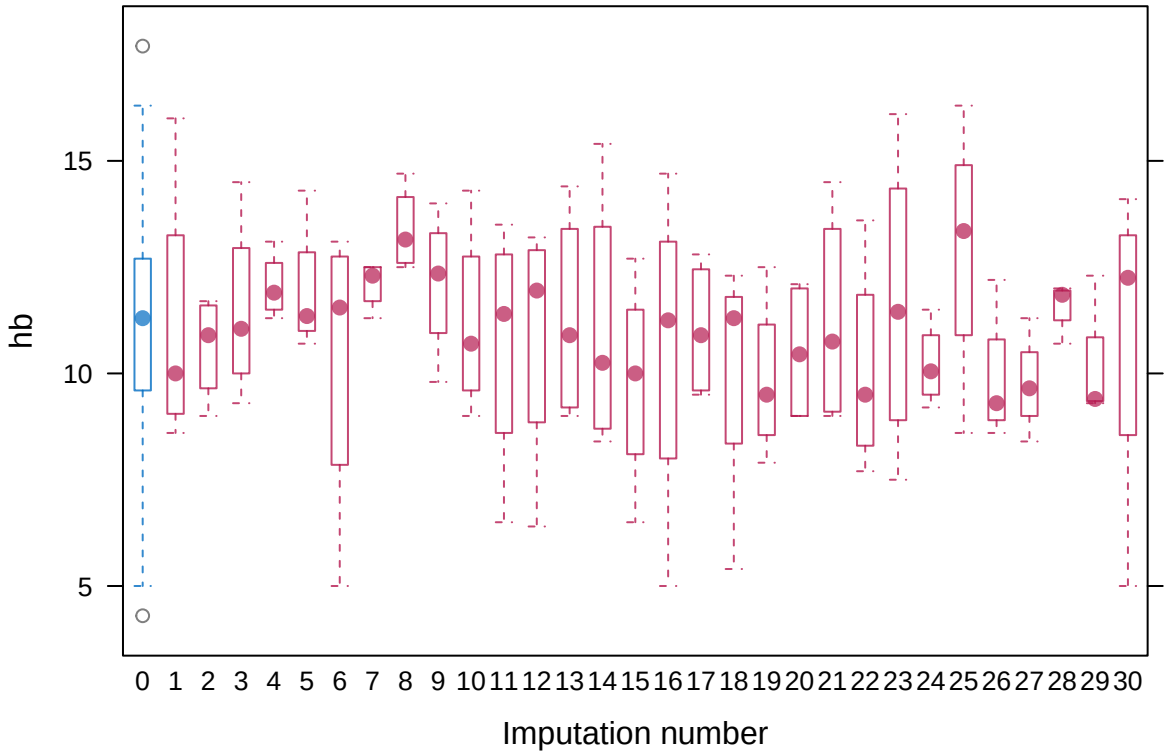

bl_age
ptretx_bmi
time_culture_pos
Imputation number




```
with(complete(imp1_d2, "long", include = TRUE), {
    boxplot(hb ~ .imp, main = "Hb across imputations")})
```




```
bwplot(imp1_d2, hb~.imp)
```



A more refined diagnostic tool that aims to compare the distributions of observed and imputed data conditional on the missingness probability. The idea is that under MAR the conditional distributions should be similar if the assumed model for creating multiple imputations has a good fit. These statements first model the probability of each record being incomplete as a function of all variables in each imputed dataset. The probabilities (propensities) are then averaged over the imputed datasets to obtain stability.

Assess proportion btw imputed and observed (categorical)

```
imp1_d2_long_include <- complete(imp1_d2,"long", include = T) %>% # change to long data
  mutate(imputed=.imp>0,
         imputed= factor(imputed,
                         levels=c(F,T),
                         labels=c("Observed", "Imputed")))

imp1_d2_long_not_include <- complete(imp1_d2,"long", include = F)
```

```
# #MTB load
prop.table(table(imp1_d2_long_include$MTB_load,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##      Observed   Imputed
##   1 0.1257485 0.1305720
##   2 0.1676647 0.1684418
##   3 0.2335329 0.2319527
##   4 0.4730539 0.4690335
```

```r
# alcohol
prop.table(table(imp1_d2_long_include$alcohol_83d0af_v2_v2,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##       Observed    Imputed
##   1 0.40993789 0.41005917
##   2 0.28571429 0.28639053
##   3 0.21739130 0.21637081
##   4 0.08695652 0.08717949
```

```r
 #smoke
prop.table(table(imp1_d2_long_include$smoker_5c21df_v2_v2,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##      Observed   Imputed
##   0 0.4785276 0.4816568
##   1 0.5214724 0.5183432
```

```r
# living alone
prop.table(table(imp1_d2_long_include$living_alone_37eb74_v2_v2,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##      Observed   Imputed
##   0 0.8834356 0.8836292
##   1 0.1165644 0.1163708
```

```r
# resistance pattern
prop.table(table(imp1_d2_long_include$resistance_pattern,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##      Observed   Imputed
##   1 0.5454545 0.5422091
##   2 0.3393939 0.3422091
##   3 0.1151515 0.1155819
```

```r
#education
prop.table(table(imp1_d2_long_include$ses_education_level,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##      Observed   Imputed
##   1 0.1304348 0.1368836
##   2 0.6304348 0.6159763
##   3 0.2391304 0.2471400
```

```r
# qol
prop.table(table(imp1_d2_long_include$qol_usual_activity,
                 imp1_d2_long_include$imputed),
           margin = 2)
```

```
##
##       Observed    Imputed
##   1 0.59523810 0.59368836
##   2 0.24404762 0.24418146
##   3 0.10119048 0.10197239
##   4 0.04761905 0.04812623
##   5 0.01190476 0.01203156
```

4. Categorize continuous variables

```r
tertile_2 <- quantile(imp1_d2_long_not_include$ses_income_before, probs = c(1/3, 2/3))
tertile_2
```

```
## 33.33333% 66.66667%
##      1800      3780
```

Categorize variable

```r
imp1_d2_long_include_cat_3 <- imp1_d2_long_include %>%
  mutate(
    # Age binary
    bl_age = factor(ifelse(bl_age >= 45, 1, 0), levels = c(0, 1)),

    # BMI: Underweight vs Non-underweight
    ptretx_bmi = cut(ptretx_bmi,
                     breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
                     labels = c(1, 0, 0, 0),  # 1: underweight, 0: normal/overweight/obese
                     right = FALSE),
    ptretx_bmi = factor(ptretx_bmi, levels = c(0, 1)),

    # Income tertiles
    ses_income_before = cut(ses_income_before,
                            breaks = c(-Inf, tertile_2[1], tertile_2[2], Inf),
                            labels = c(0, 1, 2),
                            right = TRUE),
    ses_income_before = factor(ses_income_before, levels = c(0, 1, 2)),

    # Hemoglobin: No anemia / Mild / Moderate-Severe
    hb = case_when(
      pretx_sex == 2 & hb >= 13.0  ~ 0,
      pretx_sex == 2 & hb >= 11.0 & hb < 13.0 ~ 1,
      pretx_sex == 2 & hb < 11.0 ~ 2,
      pretx_sex == 1 & hb >= 12.0 ~ 0,
      pretx_sex == 1 & hb >= 11.0 & hb < 12.0 ~ 1,
```

```
      pretx_sex == 1 & hb < 11.0 ~ 2,
      TRUE ~ NA_real_
    ),
    hb = factor(hb, levels = c(0, 1, 2)),

    # MTB load: low/very low, medium, high
    MTB_load = factor(case_when(
      MTB_load %in% c(1, 2) ~ 0,
      MTB_load == 3 ~ 1,
      MTB_load == 4 ~ 2
    ), levels = c(0, 1, 2)),

    # QoL: usual activity collapse
    qol_usual_activity = factor(case_when(
      qol_usual_activity == 1 ~ 0,
      qol_usual_activity %in% c(2, 3, 4, 5) ~ 1
    ), levels = c(0, 1)),

    # Resistance pattern: MonoDR vs MDR/(pre)XDR
    resistance_pattern = factor(case_when(
      resistance_pattern == 1 ~ 0,
      resistance_pattern %in% c(2, 3) ~ 1
    ), levels = c(0, 1))
  )
```

summarise data

```
tab1_imputed_3 <- CreateTableOne(data = imp1_d2_long_include_cat_3 %>% filter(.imp==1),strata = "late_cu
#print(tab1_imputed_3, showAllLevels = T)
```

change into mids

```
imp1_d2_long_include_cat_3_red <- imp1_d2_long_include_cat_3 %>% select(-imputed)

mids_3rd <- as.mids(imp1_d2_long_include_cat_3_red)
```

5. Build prediction model

5.1. Univariable analysis

```
# fit null model
null.model_3 <- with(mids_3rd, glm(late_culture_conversion ~ 1, family = binomial))

# Hemoglobin (HB)
uni.hb_3 <- with(mids_3rd, glm(late_culture_conversion ~ hb, family = binomial))
summary(pool(uni.hb_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term  estimate     2.5 %   97.5 %
## 1 (Intercept) 0.7239252 0.3991027 1.313115
## 2         hb1 1.7622079 0.7645363 4.061778
## 3         hb2 2.7095608 1.2538005 5.855573
```

```r
anova(null.model_3, uni.hb_3, method = "D3")
```

```
##    test statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1  3.350545   2 88127.74   166 0.0350697 0.02450735
```

```r
anova(null.model_3, uni.hb_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom    p.value        riv
##  2 ~~ 1  3.266454   2 163.6243   166 0.04064291 0.02428414
```

```r
# MTB Load
uni.mtb_load_3 <- with(mids_3rd, glm(late_culture_conversion ~ MTB_load, family = binomial))
summary(pool(uni.mtb_load_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5
```

```
##          term  estimate     2.5 %   97.5 %
## 1 (Intercept) 0.6478059 0.3661733 1.146049
## 2   MTB_load1 2.2226553 0.9394267 5.258735
## 3   MTB_load2 3.1152063 1.4838846 6.539936
```

```r
anova(null.model_3, uni.mtb_load_3, method = "D3")
```

```
##    test statistic df1    df2 dfcom     p.value         riv
##  2 ~~ 1  4.793988   2 837819   166 0.008279601 0.007814183
```

```r
anova(null.model_3, uni.mtb_load_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom    p.value         riv
##  2 ~~ 1  4.634039   2 163.9703   166 0.01102258 0.007774104
```

```r
#  Sex
uni.sex_3 <- with(mids_3rd, glm(late_culture_conversion ~ pretx_sex, family = binomial))
summary(pool(uni.sex_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term estimate     2.5 %   97.5 %
## 1 (Intercept) 1.190476 0.6635891 2.135710
## 2  pretx_sex2 1.146923 0.5772140 2.278934
```

```r
anova(null.model_3, uni.sex_3, method = "D3")
```

```
##    test statistic df1 df2 dfcom   p.value riv
##  2 ~~ 1 0.1550975   1 Inf   167 0.6937109   0
```

```r
anova(null.model_3, uni.sex_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom  p.value riv
##  2 ~~ 1 0.1553875   1 165.0353   167 0.693948   0
```

```r
# HIV Status
uni.hiv_3 <- with(mids_3rd, glm(late_culture_conversion ~ pretx_hiv, family = binomial))
summary(pool(uni.hiv_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term estimate     2.5 %   97.5 %
## 1 (Intercept) 1.142857 0.6856387 1.904972
## 2  pretx_hiv1 1.244444 0.6566992 2.358221
```

```r
anova(null.model_3, uni.hiv_3, method = "D3")
```

```
##    test statistic df1 df2 dfcom   p.value riv
##  2 ~~ 1 0.4558578   1 Inf   167 0.4995662   0
```

```r
anova(null.model_3, uni.hiv_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom  p.value riv
##  2 ~~ 1  0.456295   1 165.0353   167 0.500306   0
```

```r
# Usual Activity
uni.usual.activity_3 <- with(mids_3rd, glm(late_culture_conversion ~ qol_usual_activity, family = binomi
summary(pool(uni.usual.activity_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %"
```

```
##                  term  estimate     2.5 %   97.5 %
## 1         (Intercept) 0.9294432 0.6261402 1.379666
## 2 qol_usual_activity1 2.4420318 1.2727005 4.685721
```

```r
anova(null.model_3, uni.usual.activity_3, method = "D3")
```

```
##    test statistic df1     df2 dfcom     p.value         riv
##  2 ~~ 1  7.598904   1 1514347   167 0.005840449 0.003798323
```

```r
anova(null.model_3, uni.usual.activity_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom     p.value         riv
##  2 ~~ 1  7.317632   1 165.0003   167 0.007544944 0.003801323
```

```r
# Income
uni.income_3 <- with(mids_3rd, glm(late_culture_conversion ~ ses_income_before, family = binomial))
summary(pool(uni.income_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %
```

```
##                 term estimate     2.5 %   97.5 %
## 1        (Intercept) 1.110116 0.6317161 1.950808
## 2 ses_income_before1 1.256022 0.5579365 2.827548
## 3 ses_income_before2 1.326315 0.5774765 3.046205
```

```r
anova(null.model_3, uni.income_3, method = "D3")
```

```
##    test statistic df1      df2 dfcom   p.value       riv
##  2 ~~ 1 0.2609915   2 2261.865   166 0.7703107 0.1766327
```

```r
anova(null.model_3, uni.income_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value       riv
##  2 ~~ 1 0.2633689   2 151.7944   166 0.7688088 0.1763916
```

```r
# Education Level
uni.edu_3 <- with(mids_3rd, glm(late_culture_conversion ~ ses_education_level, family = binomial))
summary(pool(uni.edu_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##                   term  estimate     2.5 %   97.5 %
## 1          (Intercept) 1.7468993 0.6956909 4.386513
## 2 ses_education_level2 0.7626673 0.2742483 2.120930
## 3 ses_education_level3 0.6270041 0.2011986 1.953960
```

```r
anova(null.model_3, uni.edu_3, method = "D3")
```

```
##    test statistic df1      df2 dfcom  p.value     riv
##  2 ~~ 1 0.3402885   2 2745.209   166 0.711595 0.15764
```

```r
anova(null.model_3, uni.edu_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom p.value       riv
##  2 ~~ 1   0.33511   2 153.8999   166 0.71578 0.1563223
```

```r
# Living Alone
uni.living.alone_3 <- with(mids_3rd, glm(late_culture_conversion ~ living_alone_37eb74_v2_v2, family = b
summary(pool(uni.living.alone_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "
```

```
##                          term estimate     2.5 %   97.5 %
## 1                 (Intercept) 1.304533 0.9408678 1.808764
## 2 living_alone_37eb74_v2_v21 1.072335 0.4026501 2.855836
```

```r
anova(null.model_3, uni.living.alone_3, method = "D3")
```

```
##    test  statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1 0.01969885   1 15238.84   167 0.8883833 0.03930755
```

```r
anova(null.model_3, uni.living.alone_3, method = "D1")
```

```
##    test  statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1 0.01983083   1 163.1681   167 0.8881843 0.03781709
```

```r
# Smoking
uni.smoke_3 <- with(mids_3rd, glm(late_culture_conversion ~ smoker_5c21df_v2_v2, family = binomial))
summary(pool(uni.smoke_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %
```

```
##                   term  estimate     2.5 %   97.5 %
## 1          (Intercept) 1.3550664 0.8664767 2.119162
## 2 smoker_5c21df_v2_v21 0.9441619 0.5051785 1.764607
```

```r
anova(null.model_3, uni.smoke_3, method = "D3")
```

```
##    test statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1 0.0329065   1 17116.11   167 0.8560546 0.03700065
```

```r
anova(null.model_3, uni.smoke_3, method = "D1")
```

```
##    test  statistic df1      df2 dfcom   p.value       riv
##  2 ~~ 1 0.03292855   1 163.2414   167 0.8562299 0.0369968
```

```r
# Alcohol
uni.alcohol_3 <- with(mids_3rd, glm(late_culture_conversion ~ alcohol_83d0af_v2_v2, family = binomial))
summary(pool(uni.alcohol_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5
```

```
##                    term  estimate     2.5 %   97.5 %
## 1           (Intercept) 1.5574255 0.9527082 2.545978
## 2 alcohol_83d0af_v2_v22 0.7150516 0.3340234 1.530728
## 3 alcohol_83d0af_v2_v23 0.7640377 0.3321338 1.757586
## 4 alcohol_83d0af_v2_v24 0.8601067 0.2633511 2.809115
```

```r
anova(null.model_3, uni.alcohol_3, method = "D3")
```

```
##    test statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1  0.286865   3 38922.48   165 0.8349248 0.04730366
```

```r
anova(null.model_3, uni.alcohol_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1 0.2857165   3 162.1748   165 0.8356651 0.04608936
```

```r
# Prior TB Treatment
uni.prior_tb_treatment_3 <- with(mids_3rd, glm(late_culture_conversion ~ prettx_prevtbtx, family = binor
summary(pool(uni.prior_tb_treatment_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5
```

```
##               term estimate     2.5 %   97.5 %
## 1      (Intercept) 1.063830 0.7122934 1.588859
## 2 prettx_prevtbtx1 1.663077 0.8866549 3.119393
```

```r
anova(null.model_3, uni.prior_tb_treatment_3, method = "D3")
```

```
##    test statistic df1 df2 dfcom   p.value riv
##  2 ~~ 1  2.582187   1 Inf   167 0.1080722   0
```

```r
anova(null.model_3, uni.prior_tb_treatment_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value riv
##  2 ~~ 1  2.549779   1 165.0353   167 0.1122227   0
```

```r
# BMI
uni.bmi_3 <- with(mids_3rd, glm(late_culture_conversion ~ ptretx_bmi, family = binomial))
summary(pool(uni.bmi_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term  estimate     2.5 %   97.5 %
## 1 (Intercept) 0.8378378 0.5180524 1.355022
## 2 ptretx_bmi1 2.1550179 1.1454877 4.054258
```

```r
anova(null.model_3, uni.bmi_3, method = "D3")
```

```
##    test statistic df1 df2 dfcom    p.value riv
##  2 ~~ 1  5.835283   1 Inf   167 0.01570788   0
```

```r
anova(null.model_3, uni.bmi_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom    p.value riv
##  2 ~~ 1  5.754318   1 165.0353   167 0.01756311   0
```

```r
#  Age
uni.age_3 <- with(mids_3rd, glm(late_culture_conversion ~ bl_age, family = binomial))
summary(pool(uni.age_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term estimate     2.5 %   97.5 %
## 1 (Intercept) 1.134615 0.7793729 1.651779
## 2     bl_age1 1.552865 0.8047680 2.996380
```

```r
anova(null.model_3, uni.age_3, method = "D3")
```

```
##    test statistic df1 df2 dfcom   p.value riv
##  2 ~~ 1  1.773374   1 Inf   167 0.1829651   0
```

```r
anova(null.model_3, uni.age_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value riv
##  2 ~~ 1  1.747691   1 165.0353   167 0.1879967   0
```

```r
#  Resistance Pattern
uni.resistance_3 <- with(mids_3rd, glm(late_culture_conversion ~ resistance_pattern, family = binomial))
summary(pool(uni.resistance_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97
```

```
##                  term estimate     2.5 %   97.5 %
## 1         (Intercept) 1.173285 0.7731832 1.780428
## 2 resistance_pattern1 1.286676 0.6861356 2.412838
```

```r
anova(null.model_3, uni.resistance_3, method = "D3")
```

```
##    test statistic df1     df2 dfcom   p.value        riv
##  2 ~~ 1 0.6283847   1 18763.2   167 0.4279585 0.03527602
```

```r
anova(null.model_3, uni.resistance_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value        riv
##  2 ~~ 1 0.6269418   1 163.3922   167 0.4296273 0.03525935
```

```r
#  Diabetes Mellitus (DM)
uni.dm_3 <- with(mids_3rd, glm(late_culture_conversion ~ tretx_dm, family = binomial))
summary(pool(uni.dm_3), exponentiate = TRUE, conf.int = TRUE)[c("term", "estimate", "2.5 %", "97.5 %")]
```

```
##          term estimate     2.5 %   97.5 %
## 1 (Intercept) 1.308824 0.9522911 1.798840
## 2   tretx_dm1 1.069663 0.3224804 3.548057
```

```r
anova(null.model_3, uni.dm_3, method = "D3")
```

```
##    test  statistic df1          df2 dfcom   p.value           riv
##  2 ~~ 1 0.01233493   1 2.347706e+28   167 0.9115666 -3.038183e-14
```

```r
anova(null.model_3, uni.dm_3, method = "D1")
```

```
##    test statistic df1      df2 dfcom   p.value riv
##  2 ~~ 1 0.0122971   1 165.0353   167 0.9118365   0
```

5.2. Multivariable selection

There are 73 events, -> maximum 7 predictors for multivariable model after univariable analysis, 7 variables: MTB load, usual activity/ qol, prior TB treatment, malnutrition/ bmi, age, anemia/ hb, (resistance) are chosen => standard multiple regression model can be used (p<0.25)

Stepwise backward eliminations are conducted with 30 imputed dataset, models comparision based on AIC. Variables appeared in at least half of models will be chosen. Stepwise AIC across multiple imputations requires running stepAIC() on each imputed dataset individually and then pooling or summarizing the results. This method allows you to identify which variables are robust across the imputations.

- (majority) backward elimination: variables occurring > 50% in the final models were chosen

```r
scope <- list(upper = ~ MTB_load + hb + ptretx_bmi + qol_usual_activity + bl_age +  prettx_prevtbtx + r
              lower = ~1)
expr <- expression(f1 <- glm(late_culture_conversion ~ MTB_load + hb + ptretx_bmi + qol_usual_activity
                             family = binomial),
                   f2 <- step(f1, scope = scope, direction = "backward",trace =0))

# Apply the model to the imputed data
fit_3 <- with(mids_3rd, expr)
```

```r
formulas_3 <- lapply(fit_3$analyses, formula)
terms_3 <- lapply(formulas_3, terms)
votes_3 <- unlist(lapply(terms_3, labels))
table(votes_3)
```

```
## votes_3
##             bl_age                 hb          MTB_load    prettx_prevtbtx
##                 30                 23                30                 30
##         ptretx_bmi qol_usual_activity
##                 30                 30
```

hb, MTB_load, prettx_prevtbtx, ptretx_bmi, qol_usual_activity appear in all models. bl_age in 29 models, resistance_pattern in 1 models

- (D1, wald test) backward elimination , choose p-value thresshold = 0.15

```
fit_pool_3_after_majority <- with(mids_3rd, glm(late_culture_conversion~ MTB_load+ ptretx_bmi + qol_usua

fit_pool_3_not_age <- with(mids_3rd, glm(late_culture_conversion~ MTB_load+ ptretx_bmi + qol_usual_activ

D1(fit_pool_3_after_majority, fit_pool_3_not_age)
```

```
##    test statistic df1      df2 dfcom   p.value         riv
##  1 ~~ 2   2.64999   1 158.0257   160 0.1055427 0.001742126
```

```
D3(fit_pool_3_after_majority, fit_pool_3_not_age)
```

```
##    test statistic df1     df2 dfcom    p.value        riv
##  1 ~~ 2  2.713474   1 7063538   160 0.09950414 0.00175486
```

p-value = 0.119 -> keep age

5.3. Detect multicollinearity with GVIF

```
vif_model_3rd <- vif(fit_pool_3_after_majority$analyses[[1]])
# Display the VIF/GVIF values
vif_model_3rd
```

```
##           MTB_load1           MTB_load2         ptretx_bmi1 qol_usual_activity1
##            1.407067            1.387633            1.045873            1.085309
##             bl_age1                 hb1                 hb2    prettx_prevtbtx1
##            1.061347            1.543770            1.539190            1.061703
```

VIF for continuous or binary predictors. GVIF for categorical predictors with more than two levels, along with its power-transformed value GVIF^(1/(2*p*)). *Interpretation of GVIF The interpretation of GVIF^(1/(2*p*)) is similar to VIF: Values close to 1 suggest low multicollinearity. Values between 1 and 5 suggest moderate multicollinearity. Values above 5 or 10 may indicate high multicollinearity.

6. Assess model performance

6.1. Predicted accuracy

The predictive accuracy of the final model was checked using discrimination (AUC) and calibration (calibration graphs and Hosmer–Lemeshow goodness of fit test) parameters. The Hosmer–Lemeshow goodness of

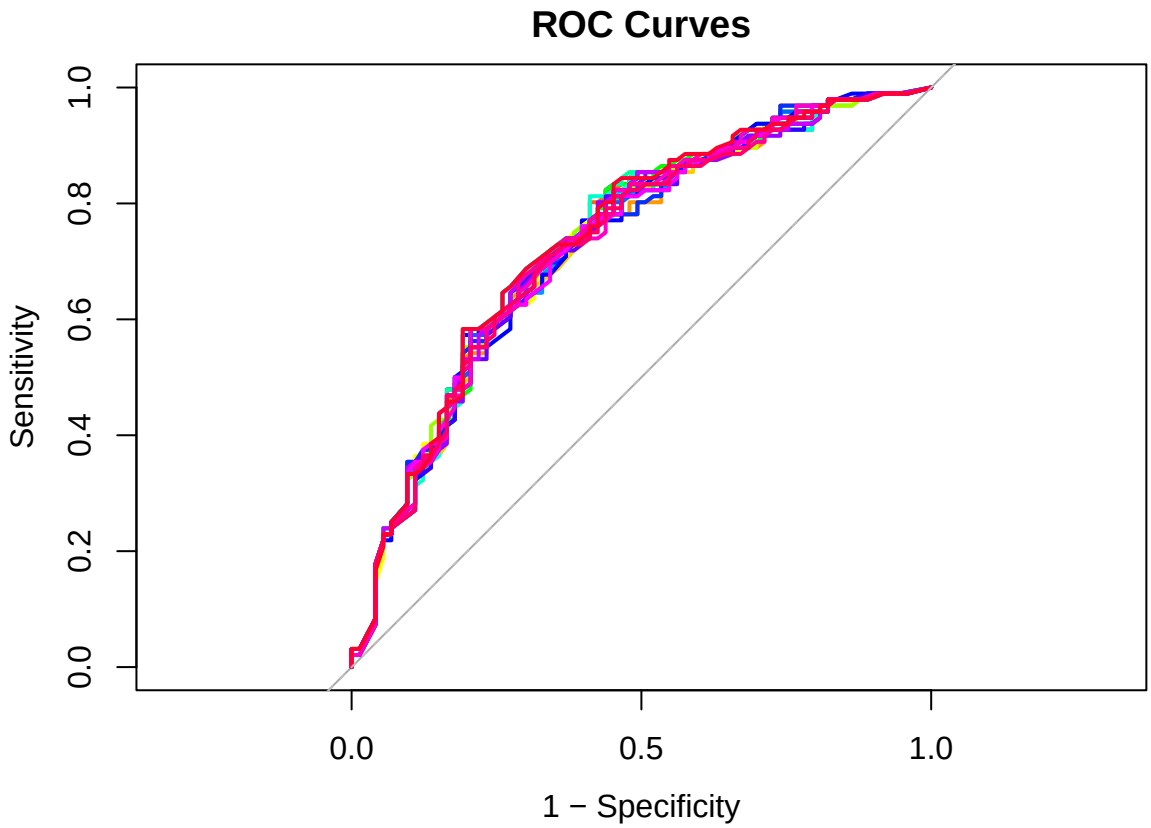
fit test with a p-value greater than 0.05 indicates good calibration, which means that the probability of late culture conversion estimated by the model is similar to the observed probability. An AUC of 0.5 indicates no discrimination ability, while an AUC of 1 indicates perfect discrimination.

AUC across 30 imputed dataset

```
# Get the predicted probabilities for each patient in all imputed dataset
predicted_probs_list_3rd <- lapply(fit_pool_3_after_majority$analyses, function(model) {
  predict(model, type = "response")
})
range(predicted_probs_list_3rd)
```

```
## [1] 0.1053515 0.9201646
```

```
# Plot only the ROC curves (no extra output)
plot.roc(roc_list[[1]], col = "blue", main = "ROC Curves",
         print.auc = FALSE, legacy.axes = TRUE)

for (i in 2:30) {
  plot.roc(roc_list[[i]], col = rainbow(30)[i], add = TRUE, print.auc = FALSE)
}
# Add diagonal line
abline(a = 1, b = -1, lty = 2, col = "gray")
```



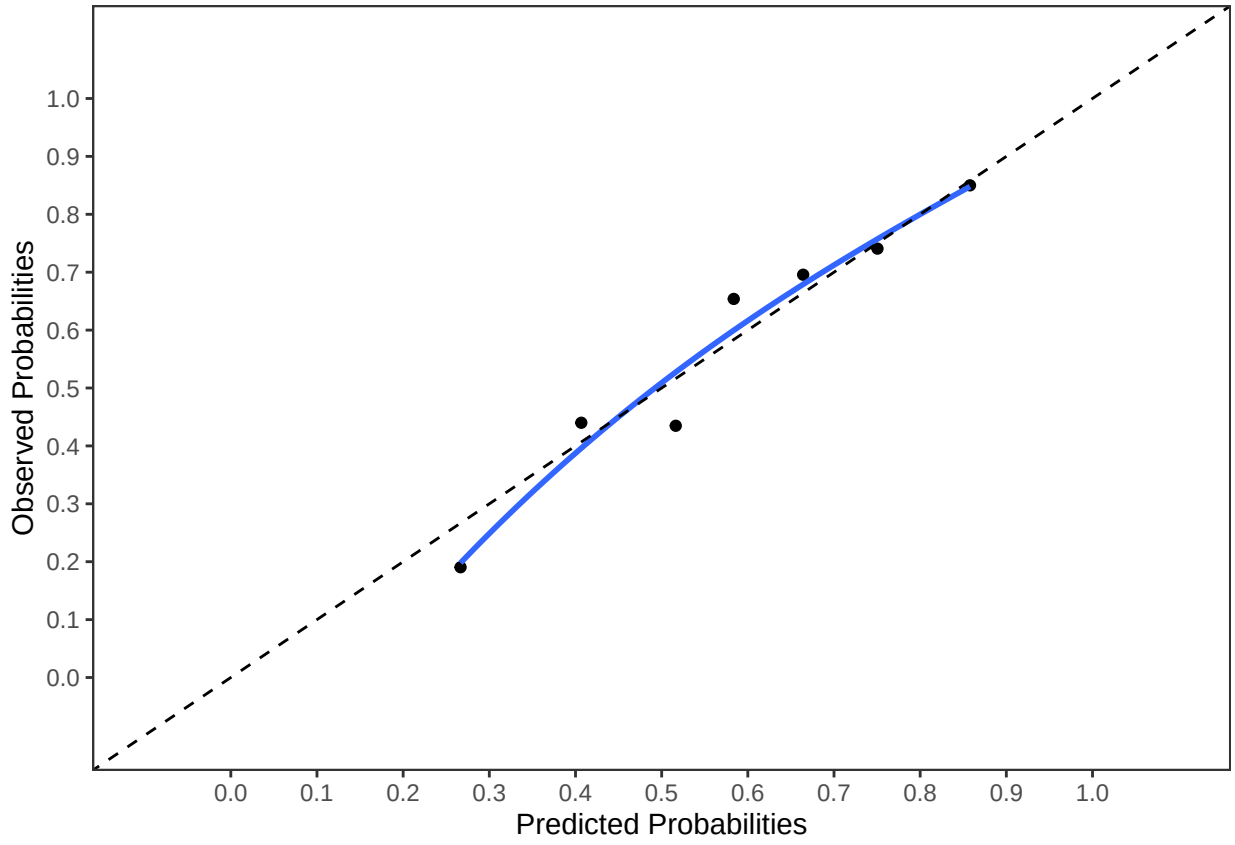
```r
# Calculate the Youden Index and the corresponding threshold for each ROC curve
youden_results <- lapply(roc_list, function(roc_obj) {
  # Get the optimal threshold that maximizes the Youden Index (sensitivity + specificity - 1)
  coords(roc_obj, "best", ret = c("threshold", "sensitivity", "specificity", "youden"),
         transpose = FALSE)
})

# Extract thresholds and Youden Indexes
thresholds <- sapply(youden_results, function(res) res$threshold)
youden_indexes <- sapply(youden_results, function(res) res$youden)

# Print the thresholds and Youden Indexes
data.frame(Threshold = thresholds, YoudenIndex = youden_indexes)
```

```
##    Threshold YoudenIndex
## 1  0.5695512    1.386130
## 2  0.5616390    1.372432
## 3  0.5136308    1.391124
## 4  0.5893414    1.361444
## 5  0.6130654    1.377854
## 6  0.6170288    1.377854
## 7  0.5746708    1.378995
## 8  0.5779420    1.375713
## 9  0.6238532    1.377854
## 10 0.5668043    1.371861
## 11 0.4999295    1.384561
## 12 0.6206914    1.377854
## 13 0.5668043    1.371861
## 14 0.6196744    1.391553
## 15 0.5081678    1.401541
## 16 0.6238532    1.377854
## 17 0.6196337    1.377854
## 18 0.6123370    1.377854
## 19 0.5653514    1.372432
## 20 0.5616390    1.372432
## 21 0.5242394    1.349458
## 22 0.6182824    1.377854
## 23 0.5124947    1.374144
## 24 0.5216728    1.377426
## 25 0.5700906    1.375713
## 26 0.5434256    1.363156
## 27 0.6060698    1.377854
## 28 0.5685567    1.378995
## 29 0.5653077    1.365868
## 30 0.6247400    1.391553
```

Pooled calibration + AUC: pool_performance Pooling performance measures (AUC): aggregates performance measures from multiple imputed datasets using a method that accounts for the variability between imputations

```r
stacked_long_3rd <- complete(mids_3rd, "long", include = F)
pool_performance_3 <- pool_performance(stacked_long_3rd,
                nimp = 30,
```

```
                      impvar = ".imp",
                      formula = late_culture_conversion~ bl_age+ hb+ MTB_load+ prettx_prevtbtx +ptretx_bl
                     cal.plot=TRUE, plot.method="mean",
groups_cal=7, model_type="binomial")
```




```
pool_performance_3$ROC_pooled
```

```
##                      95% Low C-statistic 95% Up
## C-statistic (logit)  0.6483      0.7322 0.8022
```

```
pool_performance_3$R2_pooled
```

```
## [1] 0.2056982
```

```
pool_performance_3$HLtest_pooled
```

```
##         F_value     P(>F) df1      df2
## [1,] 0.2628084 0.9333774   5 1117.869
```

```
pool_performance_3$Brier_Scaled_pooled
```

```
## [1] 0.1599911
```

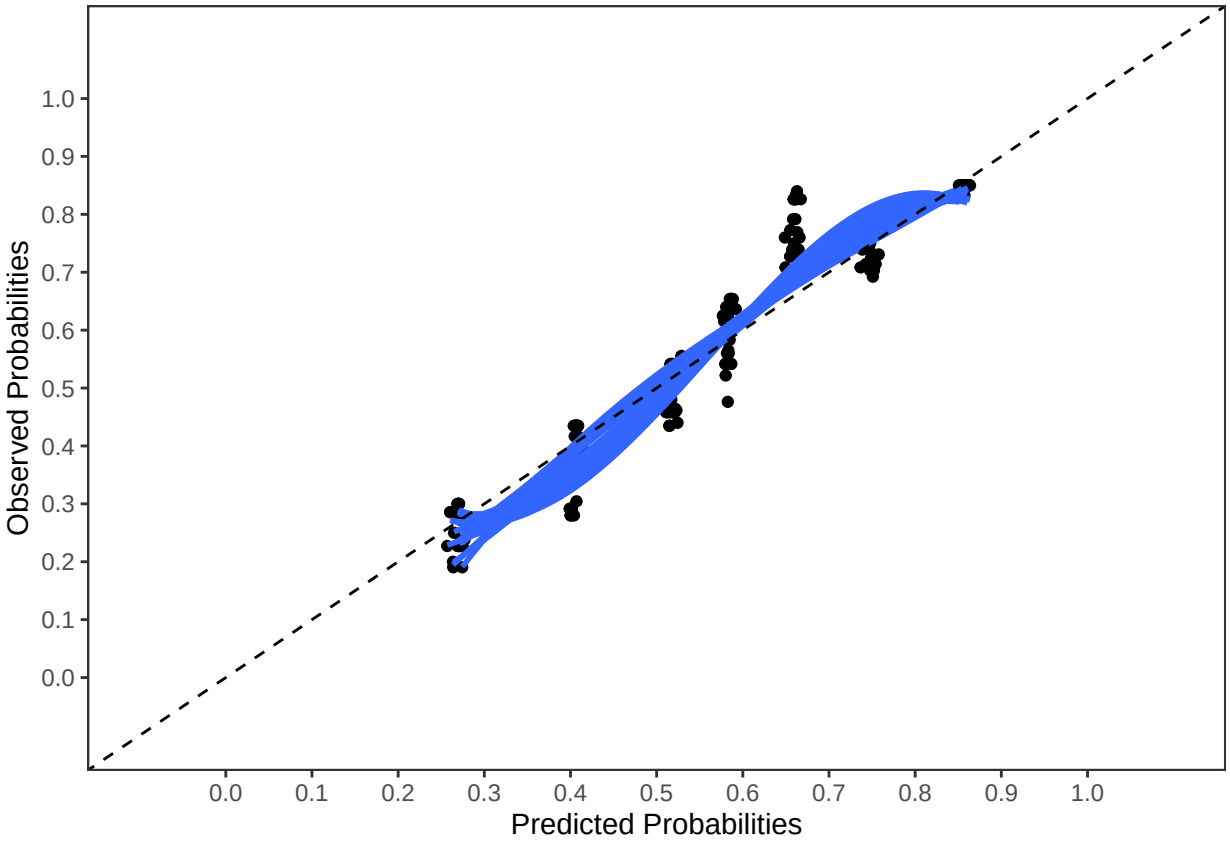
R^2 of 0.2067 suggests that about 20.67% of the variance in the outcome is explained by the model The Brier score measures overall performance—it is simply calculated as the mean squared difference between predicted probabilities and actual outcomes. The value of the Brier score estimate can range between 0 and 1. A lower Brier score value indicates a better performing model.

Hosmo Lemeshow test: to assess goodness-of-fit for logistic models It evaluates whether the observed event rates match the expected event rates across deciles of predicted probabilities.

p-value >0.05 suggests that there is no significant difference between the observed and expected frequencies in your data across different deciles of predicted probabilities. In other words, your model fits the data well according to the Hosmer-Lemeshow test.

Overlay calibration

```
cali_plot <- pool_performance(data = stacked_long_3rd, formula = late_culture_conversion~ bl_age+ hb+ M
                 nimp=30, impvar=".imp",
                 cal.plot=TRUE, plot.method="overlay",
                 groups_cal=7, model_type="binomial")
```



6.2. Internal validation: bootstraping

The model was internally validated using a bootstrap technique to estimate the degree of over-optimism of the final model when applied to a similar population. Internal validation was performed on the regression coefficient with a 95% confidence interval (CI) and the AUC of the model using 2,000 random bootstrap samples. The AUC difference between the bootstrap and the original full sample measured the optimism of the predictive model.

Ideally, we should first bootstrap and then impute. However, this strategy might be computationally difficult. Instead, we can first impute, then bootstrap, obtain optimism corrected performance measures from each imputed dataset, and finally pool these

Method cv_MI uses imputation within each cross-validation fold definition. By repeating this in several imputation runs, multiply imputed datasets are generated. Method cv_MI_RR uses multiple imputation within the cross-validation definition. MI_cv_naive, applies cross-validation within each imputed dataset. MI_boot draws for each bootstrap step the same cases in all imputed datasets. With boot_MI first bootstrap samples are drawn from the original dataset with missing values and than multiple imputation is applied. For multiple imputation the mice function from the mice package is used. It is recommended to use a minumum of 100 imputation runs for method cv_MI or 100 bootstrap samples for method boot_MI or MI_boot

Orig (original datasets), Apparent (models applied in bootstrap samples), Test (bootstrap models are applied in original datasets), Optimism (difference between apparent and test) and Corrected (original corrected for optimism).

```
#Pooled model wald test
pool_D1_model_3rd <- psfmi_lr(data = stacked_long_3rd,
         nimp = 30, # number of imputation
         impvar = ".imp",  # The variable that identifies imputation number
         formula = late_culture_conversion ~ MTB_load + hb + ptretx_bmi + qol_usual_activity + bl_age +
         method = "D1",  # Rubin's rules (D1 method)
         p.crit = 1)  # Keep all predictors (no variable selection)

internal_validate_3 <- suppressMessages(
  suppressWarnings(
    psfmi_validate(
      pobj       = pool_D1_model_3rd,
      val_method = "MI_boot",
      miceImp    = TRUE,
      int_val    = TRUE,
      nboot      = 200,  # number of bootstrap resamples
      plot.method = "mean",
      cal.plot   = TRUE,
      groups_cal = 7
    )
  )
)

#internal_validate_3$intercept_test
#internal_validate_3$res_boot$R2_app
```

```
auc_values <- internal_validate_3$res_boot$ROC_app
quantile(auc_values, 0.025)
```

```
##      2.5%
## 0.6847675
```

```
quantile(auc_values, 0.975)
```

```
##     97.5%
## 0.8261175
```

Characteristic	OR[1]	95% CI[1]	p-value
bl_age			
0	—	—	
1	1.82	0.88, 3.77	0.11
hb			
0	—	—	
1	1.97	0.79, 4.89	0.14
2	2.31	0.98, 5.44	0.056
MTB_load			
0	—	—	
1	1.82	0.71, 4.63	0.2
2	2.55	1.15, 5.65	0.021
prettx_prevtbtx			
0	—	—	
1	1.98	0.98, 3.99	0.057
ptretx_bmi			
0	—	—	
1	1.89	0.94, 3.78	0.072
qol_usual_activity			
0	—	—	
1	1.98	0.97, 4.07	0.062

[1]OR = Odds Ratio, CI = Confidence Interval

7. Risk score construction

A simplified risk score was constructed based on the hierarchy of the regression coefficients in the final model (each coefficient was divided by the smallest coefficient (bl_age =0.60) and rounded to the nearest integer). The score's predictive performance (AUC) was assessed and compared with that of the original model. Sustained culture conversion risk corresponding to each score was calculated

```
fit_pool_3_final <- with(mids_3rd, glm(late_culture_conversion~ bl_age+ hb+ MTB_load+ prettx_prevtbtx +
tbl_regression(fit_pool_3_final,exponentiate = TRUE)
```

```
# MTB_low is smallest -> weight of 1
# age1 (weight of 1)
0.5627385/ 0.5511816
```

```
## [1] 1.020967
```

```
# hb1 (weight of 1)
0.7064516/ 0.5511816
```

```
## [1] 1.281704
```

```r
# MTB load 2 (weight of 2)
0.9132189/ 0.5511816
```

```
## [1] 1.656839
```

```r
# prettx_prevtbtx (weight of 1)
0.6715238/0.5511816
```

```
## [1] 1.218335
```

```r
# bmi (weight of 1)
0.6424174/0.5511816
```

```
## [1] 1.165528
```

```r
# qol (weight of 1)
0.7490449/0.5511816
```

```
## [1] 1.35898
```

Total score = 1* age>= 45 + 1 * anemia_mild + 1* MTB_load_medium + 2 * MTB_load_high + 1* pre_TB_treatment + 1* underweight + 1* qol_some_problems (baseline: age < 45, MTB_load: low/very low, bmi: non-underweight, hb: non-anemia, qol: no problem, pre_TB_treatment: no)

Total score = 1 * bl_age1 + 1 * hb1 + 1* MTB_load1 + 2* MTB_load2 + 1* prettx_prevtbtx1 + 1* ptretx_bmi0 + 1* qol_usual_activity1 min = 0, max = 7

```r
imp1_d2_long_include_cat_3_red_score <- imp1_d2_long_include_cat_3_red %>%
  mutate(MTB_load_RC1 = if_else(MTB_load==1, 1, 0),
         MTB_load_RC2 = if_else(MTB_load==2, 1, 0),
         TotalScore = 1 * as.numeric(as.character(bl_age)) +
             1 * as.numeric(as.character(hb)) +
             1 * MTB_load_RC1 +
             2 * MTB_load_RC2 +
             1 * as.numeric(as.character(prettx_prevtbtx)) +
             1 * as.numeric(as.character(ptretx_bmi)) +
             1 * as.numeric(as.character(qol_usual_activity)))

imp1_d2_long_not_include_cat_3_red_score  <- imp1_d2_long_include_cat_3_red_score %>%
  filter(.imp!=0)
```

Calculate risk scores for stacked 30 imputed datasets (equal contribution of imputed dataset)

```r
imp1_d2_long_not_include_cat_3_red_score %>%
  mutate(TotalScore = case_when(
    TotalScore %in% 0:1 ~ "0-1",  # Collapse scores 0 and 1 into "0-1"
    TRUE ~ as.character(TotalScore)  # Keep scores 2-7 unchanged
  )) %>%
  group_by(TotalScore) %>%
  summarise(
```

```
    total_patients = n(),  # Total patients with this score
    n_late_culture_conversion = sum(late_culture_conversion == 1),  # Late culture conversion cases
    Proportion = round(n_late_culture_conversion / total_patients, 3) # Proportion
  ) %>%
  arrange(factor(TotalScore, levels = c("0-1", "2", "3", "4", "5", "6", "7")))  # Ensure correct order
```

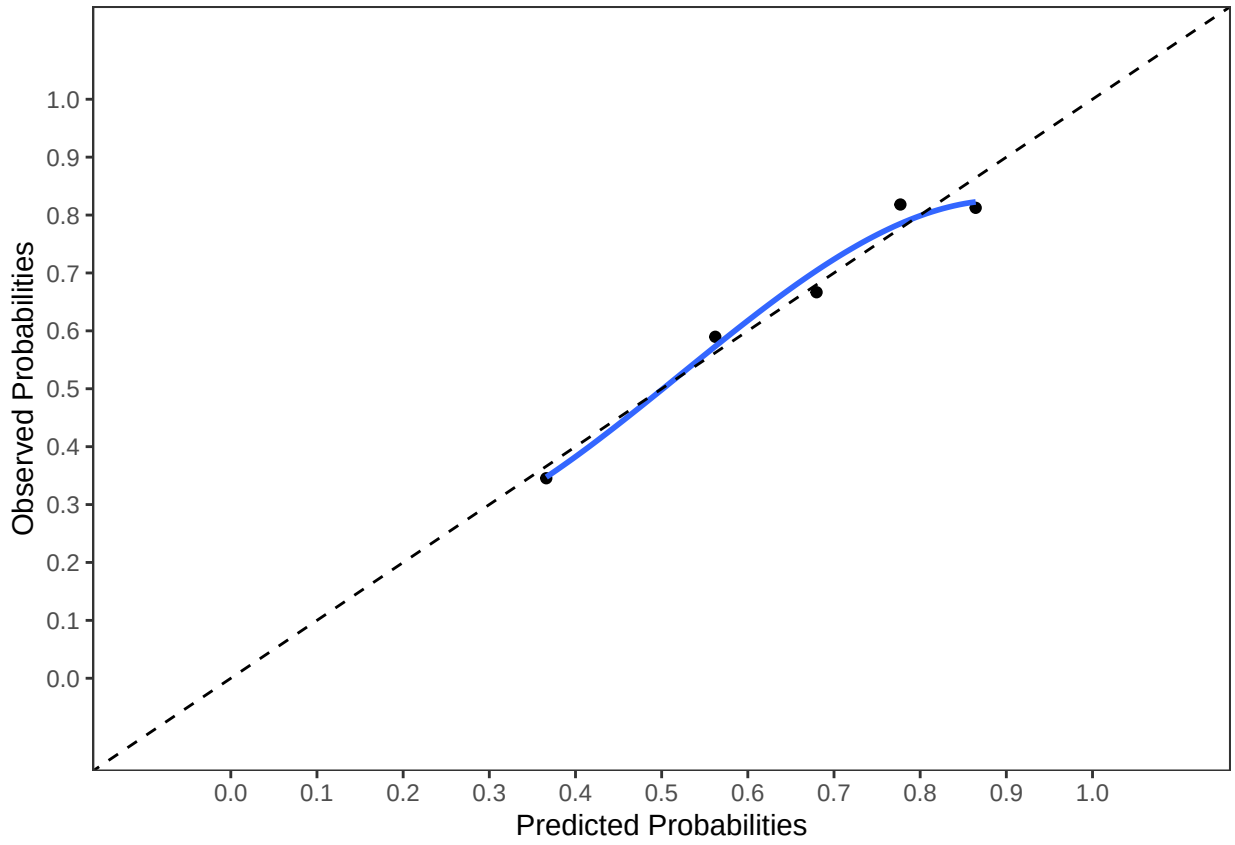
```
## # A tibble: 8 x 4
##   TotalScore total_patients n_late_culture_conversion Proportion
##   <chr>               <int>                     <int>      <dbl>
## 1 0-1                   355                        64      0.18
## 2 2                     548                       179      0.327
## 3 3                     890                       360      0.404
## 4 4                    1161                       704      0.606
## 5 5                     997                       643      0.645
## 6 6                     639                       540      0.845
## 7 7                     400                       310      0.775
## 8 8                      80                        80      1
```

```
imp1_d2_long_not_include_cat_3_red_score %>%
  filter(.imp==2) %>%
  filter(TotalScore %in% 0:7) %>%
  group_by(TotalScore) %>%  # Group by TotalScore
  summarise(
    n_total = n(),  # Total patients with this score
    n_late_conversion = sum(late_culture_conversion == 1),  # Patients with late culture conversion
    proportion_late_conversion = n_late_conversion / n_total  # Proportion
  ) %>%
  arrange(TotalScore)
```

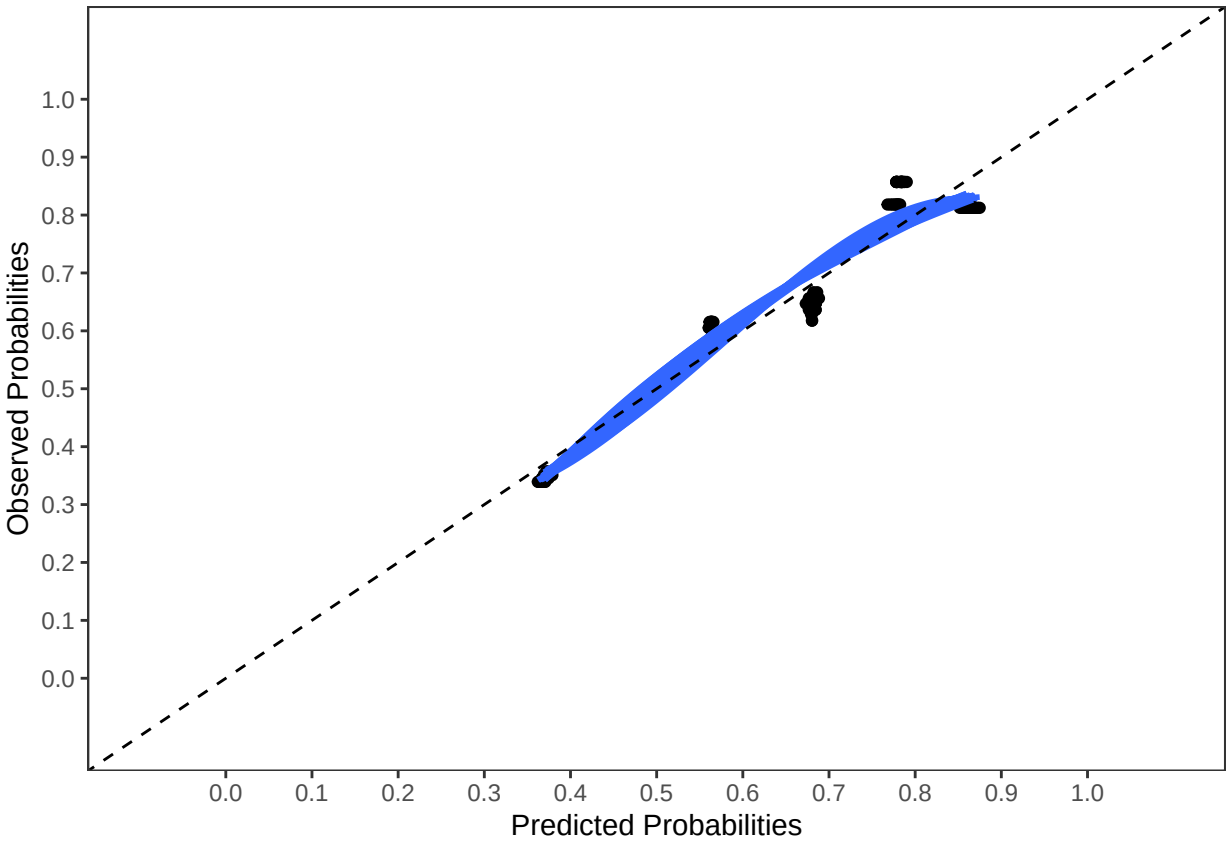
```
## # A tibble: 8 x 4
##   TotalScore n_total n_late_conversion proportion_late_conversion
##        <dbl>   <int>             <int>                      <dbl>
## 1          0       4                 1                      0.25
## 2          1       8                 1                      0.125
## 3          2      18                 6                      0.333
## 4          3      30                12                      0.4
## 5          4      38                23                      0.605
## 6          5      34                22                      0.647
## 7          6      21                18                      0.857
## 8          7      13                10                      0.769
```

7.1. Performance for risk scores

```
pool_performance_score <- pool_performance(imp1_d2_long_include_cat_3_red_score,
                nimp = 30,
                     impvar = ".imp",
                     formula = late_culture_conversion~ TotalScore,
                    cal.plot=TRUE, plot.method="mean",
 groups_cal=5, model_type="binomial")
```




```
pool_performance(imp1_d2_long_include_cat_3_red_score,
                 nimp = 30,
                      impvar = ".imp",
                      formula = late_culture_conversion~ TotalScore,
                     cal.plot=TRUE, plot.method="overlay",
 groups_cal=5, model_type="binomial")
```




```
## $ROC_pooled
##                       95% Low C-statistic 95% Up
## C-statistic (logit)  0.6382      0.7206 0.7905
##
## $coef_pooled
## (Intercept)  TotalScore
##  -1.7757380   0.5078212
##
## $R2_pooled
## [1] 0.1953933
##
## $Brier_Scaled_pooled
## [1] 0.1495607
##
## $nimp
## [1] 30
##
## $HLtest_pooled
##        F_value     P(>F) df1      df2
## [1,] 0.4651624 0.7065946   3 37521.59
##
## $model_type
## [1] "binomial"
```

group_cal: the number of risk groups for calibration

```r
pool_performance_score$ROC_pooled
```

```
##                      95% Low C-statistic 95% Up
## C-statistic (logit)  0.6382      0.7206 0.7905
```

```r
pool_performance_score$coef_pooled
```

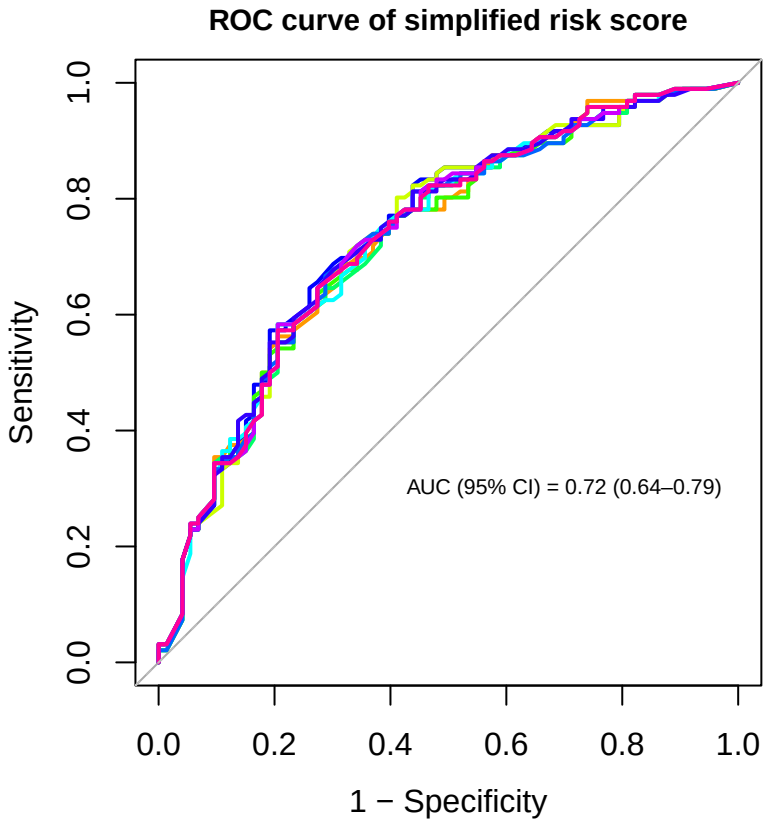
```
## (Intercept)  TotalScore
##  -1.7757380   0.5078212
```

```r
pool_performance_score$HLtest_pooled
```

```
##        F_value     P(>F) df1      df2
## [1,] 0.4651624 0.7065946   3 37521.59
```

AUC draw for risk score

```r
par(pty = "s")

# Plot the first ROC curve
plot.roc(roc_list[[1]],
         col = "blue",

         print.auc = FALSE,
         legacy.axes = TRUE,
         main = "ROC curve of simplified risk score",
         cex.main=0.9)

# Add other ROC curves
for (i in 2:30) {
  plot.roc(roc_list[[i]], col = rainbow(10)[i], add = TRUE, print.auc = FALSE)
}

# Add diagonal line
abline(a = 1, b = -1, lty = 2, col = "gray")

# Add AUC text
text(x = 0.3, y = 0.3,
     labels = "AUC (95% CI) = 0.72 (0.64-0.79)",
     cex = 0.7, font = 1)
```



7.2. Youden index

Threshold that maximizes the sum of sensitivity and specificity — i.e., the point on the ROC curve farthest from the diagonal

```
mids_3rd_score <- as.mids(imp1_d2_long_include_cat_3_red_score)

# Get predicted probabilities per imputation
pred_matrix <- sapply(1:30, function(i) {
  dat_i <- complete(mids_3rd_score, i)
  model <- glm(late_culture_conversion ~ TotalScore,
               data = dat_i, family = binomial)
  predict(model, type = "response")  # vector of predicted probabilities
})

# Get the average prediction per individual (pooled prediction)
pooled_preds <- rowMeans(pred_matrix)

# Use the observed outcome from any imputed dataset (e.g., first one)
true_outcome <- complete(mids_3rd_score, 1)$late_culture_conversion

# Calculate ROC with pooled predicted probabilities
pooled_roc <- roc(true_outcome, pooled_preds, direction = "<")
```

```
## Setting levels: control = 0, case = 1
```

```r
# Get optimal threshold using Youden Index
youden_result <- coords(pooled_roc,
                        x = "best",
                        best.method = "youden",
                        ret = c("threshold", "sensitivity", "specificity", "youden"),
                        transpose = FALSE)
print(youden_result)
```

```
##   threshold sensitivity specificity   youden
## 1 0.5365736   0.7916667   0.5479452 1.339612
```