SAT GPA data synthesis and evaluation

IPUMS International Team

1/28/2022

```
suppressPackageStartupMessages({
  library(synthpop)
  library(readr)
  library(dplyr)
  library(purrr)
  library(ggplot2)
})
```

```
## Warning: package 'synthpop' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
data_path <- "../data/satgpa.csv"
```

```
satgpa <- read_csv(
  data_path,
  col_types = cols(
    sex = col_double(),
    sat_v = col_double(),
    sat_m = col_double(),
    sat_sum = col_double(),
    hs_gpa = col_double(),
    fy_gpa = col_double()
  )
)
```

In this analysis, we will generate a synthetic dataset of information on student SAT scores and GPAs, evaluate the utility and disclosure risk of that synthetic dataset, then attempt to tune the model that generates the synthetic dataset to improve utility while maintaining low disclosure risk.

Generate initial synthetic dataset

Our initial model includes a few custom settings. First, it uses the `minnumlevels` argument to specify that variables with two or fewer unique values should be treated as factors. Second, it specifies a "passive" synthesizing method for variable `sat_sum`, which is just the sum of `sat_v` and `sat_m`. Otherwise, this initial model uses all default settings.

```r
set.seed(2022-1-25)
synth1 <- syn(
  satgpa,
  minnumlevels = 2,
  method = c(
    sex = "sample", sat_v = "cart", sat_m = "cart",
    sat_sum = "~I(sat_v + sat_m)", hs_gpa = "cart", fy_gpa = "cart"
  )
)
```

```
##
## Variable(s): sex numeric but with only 2 or fewer distinct values turned
into factor(s) for synthesis.
##
## Synthesis
## -----------
##  sex sat_v sat_m sat_sum hs_gpa fy_gpa
```

Let's confirm that `sat_sum` is always the sum of `sat_v` and `sat_m`:

```r
with(satgpa, all(sat_m + sat_v == sat_sum))
```

```
## [1] TRUE
```

```r
with(synth1$syn, all(sat_m + sat_v == sat_sum))
```

```
## [1] TRUE
```

Now let's write this dataset to a .csv file so that we can use Python tools to evaluate it as well:

```r
write_csv(synth1$syn, "../synthetic-data/satgpa-naive-synth.csv")
```

Evaluate initial synthetic dataset

Evaluate utility

First, let's use a general utility measure to assess the performance of our synthetic dataset:

```r
utility1 <- utility.gen(synth1, satgpa)
```

```
## Running 50 permutations to get NULL utilities and printing every 10th.
## synthesis 1   10 20 30 40 50
```

```r
utility1$pMSE
```

```
## [1] 0.04779752
```

The pMSE is the average squared difference from 0.5 of the probability that a given record is synthetic, when trying to distinguish between the synthetic records and the real records.

Thus, a pMSE of 0.048 indicates that the average record had a probability of 0.5 +- 0.22, in other words, about 0.28 or 0.72. That seems like okay utility, not great, but not terrible. The better of two models in the synthpop utility vignette has a pMSE of 0.0085, which translates to an average record having probability of about 0.41 or 0.59 of being synthetic. That looks significantly better than the 0.048 score, when translated to that probability scale. Thus, it seems our initial model might have room for improvement.

To decide where there might be room for improvement, let's compare the one-way tables of each variable between our two datasets:

```
compare(
  synth1,
  as.data.frame(satgpa),
  utility.stats = c("S_pMSE", "df"),
  plot = FALSE
)
```

```
##
## Comparing percentages observed with synthetic
##
##
## Selected utility measures:
##           S_pMSE df
## sex     1.296254  1
## sat_v   0.479595  4
## sat_m   1.392973  4
## sat_sum 1.094954  4
## hs_gpa  0.911514  4
## fy_gpa  0.202887  4
```

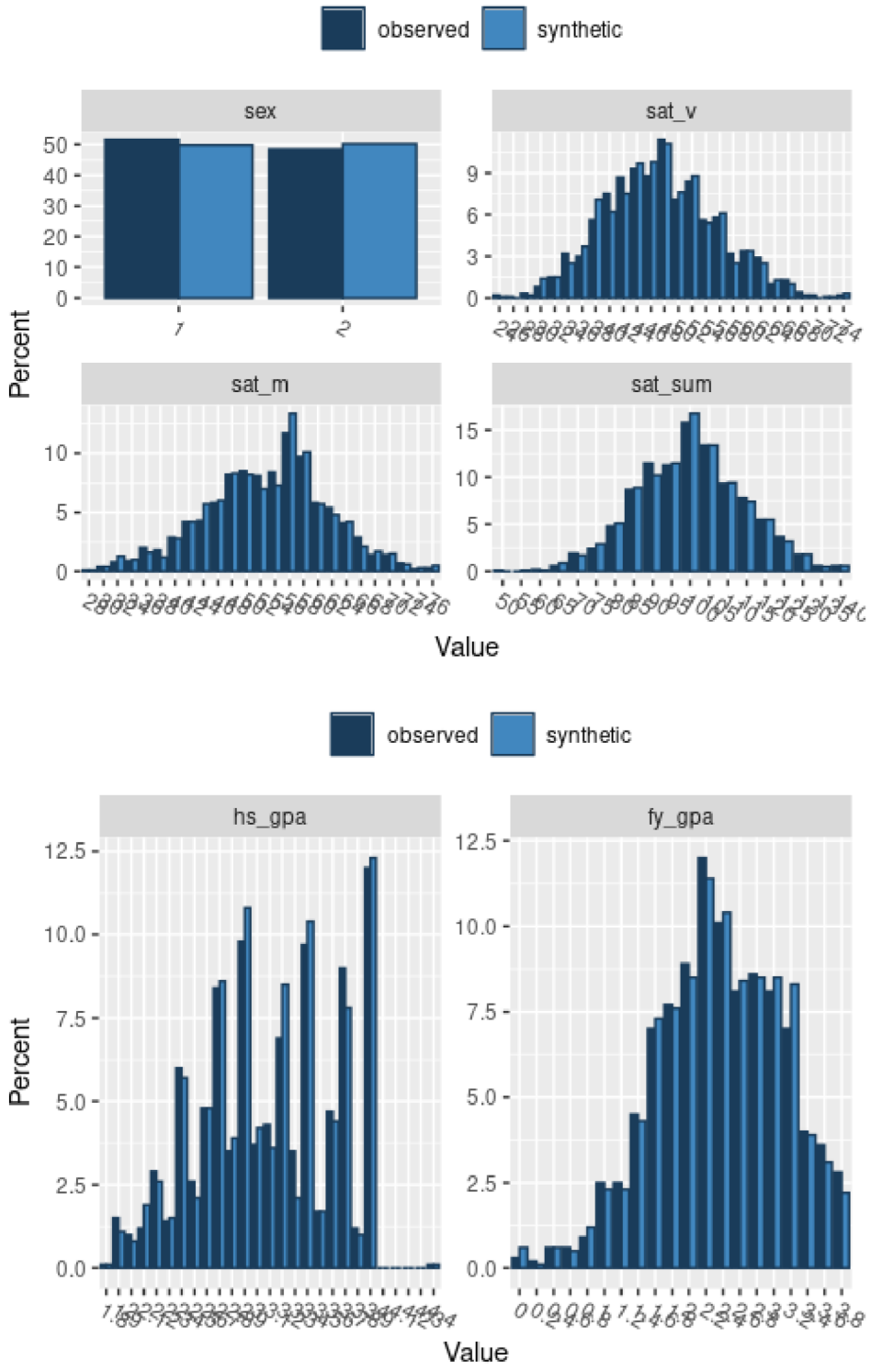
The target for the S_pMSE measure is 1.0, and the creators of the synthpop package suggest in the utility vignette that any value below 10 could be acceptable. Thus, none of these variables stands out as especially problematic, but sat_m does have the worst (highest) score. The vignette suggests moving variables with poor S_pMSE scores toward the beginning of the visit sequence, so we can try that below.

We can also plot the distributions of each variable in the original and synthetic datasets:

```
compare1 <- compare(synth1, as.data.frame(satgpa))
compare1$tab.utility
```

```
##                  pMSE    S_pMSE df
## sex     8.101588e-05 1.2962541  1
## sat_v   1.198987e-04 0.4795949  4
## sat_m   3.482432e-04 1.3929726  4
## sat_sum 2.737385e-04 1.0949541  4
## hs_gpa  2.278785e-04 0.9115141  4
## fy_gpa  5.072171e-05 0.2028868  4
```

```r
for (plt in compare1$plots) {
  print(plt)
}
```

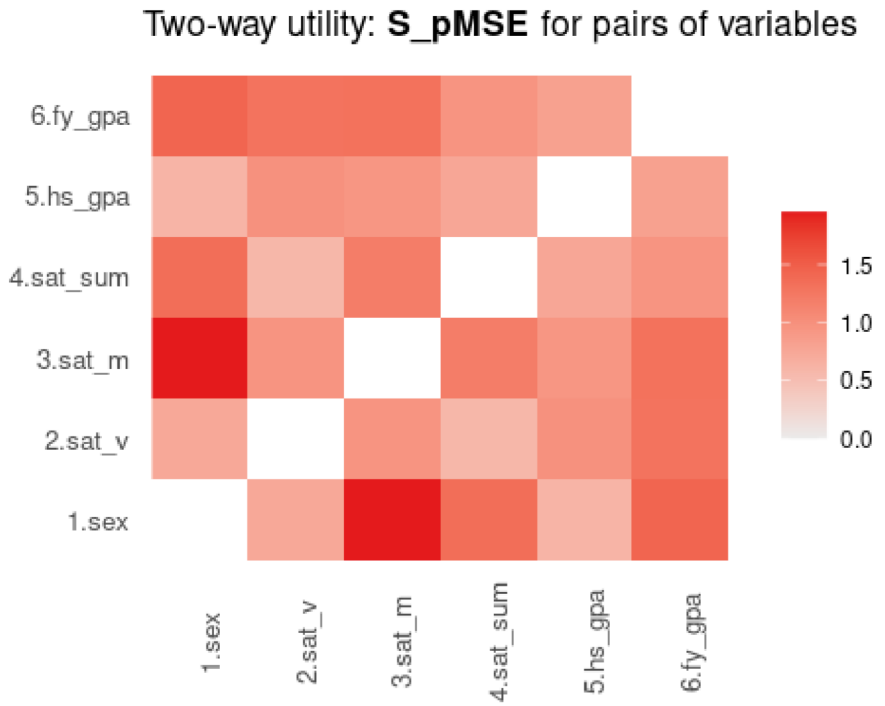


Nothing stands out in these graphs as particularly concerning.

Now let's compare all two-way tables between the original and synthetic data:

```
twoway_tabs1 <- utility.tables(
  synth1,
  as.data.frame(satgpa),
  tables = "twoway"
)
twoway_tabs1

##
## Two-way utility: S_pMSE value plotted for 15 pairs of variables.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
##    1.sex:3.sat_m   1.sex:6.fy_gpa  1.sex:4.sat_sum 3.sat_m:6.fy_gpa
2.sat_v:6.fy_gpa
##           1.9552           1.4369           1.3561           1.3185
1.3001
```



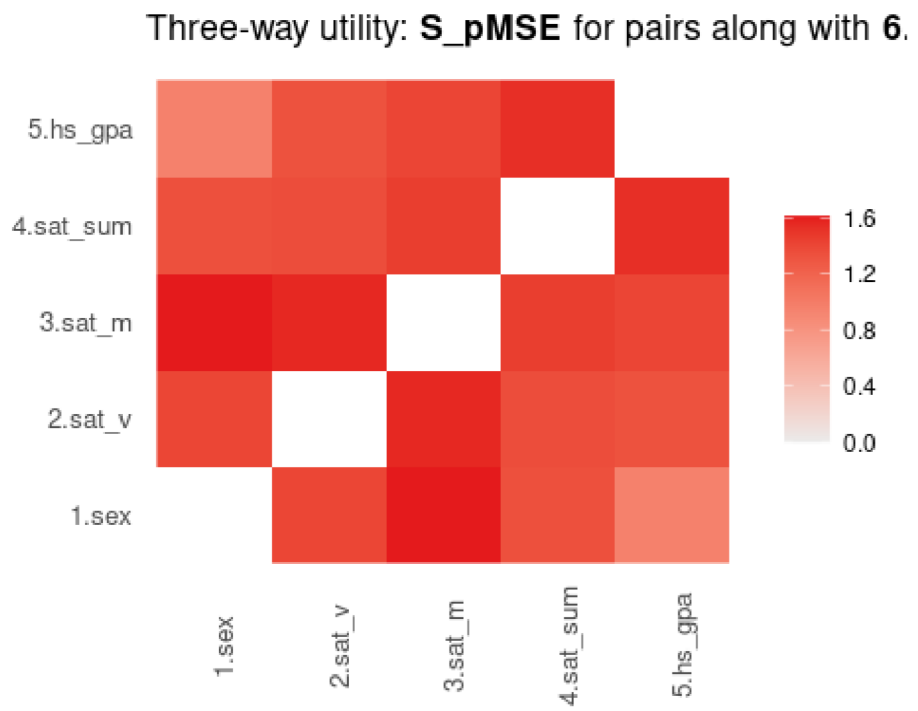

```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians  Maxima
## pMSE    0.0012  0.0020
## S_pMSE  0.9644  1.9552
## df     24.0000 24.0000
##
## For more details of all scores use print.tabs = TRUE.
```

Here, we see the greatest discrepancies for the sex by sat_m table, which again suggests that we could improve upon the synthesis of sat_m, though even for that table our S_pMSE value is not too large.

Finally, if we assume we are particularly interested in the first-year GPA variable as a key outcome, we might want to look at all three-way tables involving that variable:

```
threeway_tabs1 <- utility.tables(
  synth1,
  as.data.frame(satgpa),
  tables = "threeway",
  third.var = "fy_gpa"
)
threeway_tabs1
```

```
##
## Three-way utility (total of 20 variable combinations):
##
## Average of 3-way scores S_pMSE (ordered) for 3-way tables including each
variable.
##  6.fy_gpa   3.sat_m 4.sat_sum  5.hs_gpa     1.sex   2.sat_v
##  1.397452  1.278826  1.195979  1.183172  1.177466  1.169369
##
## Variable with highest average score, 6.fy_gpa, selected to make plots.
## To see others, set parameter 'third.var'.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
##      1.sex:3.sat_m:6.fy_gpa    2.sat_v:3.sat_m:6.fy_gpa
4.sat_sum:5.hs_gpa:6.fy_gpa  3.sat_m:4.sat_sum:6.fy_gpa
##                      1.6133                      1.5650
1.5324                      1.4553
##   3.sat_m:5.hs_gpa:6.fy_gpa
##                      1.4122
```




```
##
## Medians and maxima of selected utility measures for all tables compared
##         Medians   Maxima
## pMSE     0.0046   0.0118
## S_pMSE   1.2423   1.6133
## df      50.5000 124.0000
##
## For more details of all scores use print.tabs = TRUE.
```

Again we see that the tabulation of sex and sat_m, this time with fy_gpa, has the greatest discrepancies between original and synthetic data.

Evaluate disclosure risk

One way to evaluate disclosure risk is to find all cases that match exactly on a set of "quasi-identifiers" between the original and synthetic data, then assess the degree to which those apparent matches have the same values on other variables in the datasets. As an exercise, we will pretend that sex and SAT verbal scores act as quasi-identifiers, and assess the degree to which unique records that match on those characteristics between the original and synthetic data also match on other characteristics.

```
cellchange <- function(df1, df2, quasi, exclude_cols = NULL) {
  uniques1 <- df1 %>%
    add_count(across(all_of(quasi))) %>%
    filter(n == 1)
```

```
  uniques2 <- df2 %>%
    add_count(across(all_of(quasi))) %>%
    filter(n == 1)
  matcheduniq <- inner_join(uniques1, uniques2, by = quasi)
  allcols <- intersect(names(df1), names(df2))
  cols <- setdiff(allcols, quasi)
  cols <- setdiff(cols, exclude_cols)
  percents <- match(matcheduniq, cols)

  list(percents = percents, uniques1 = uniques1, uniques2 = uniques2,
       matched = matcheduniq)
}

match <- function(df, cols) {
  map_dfc(
    cols,
    ~tibble(!!sym(.x) := df[[paste0(.x, ".x")]] == df[[paste0(.x, ".y")]])
  ) %>%
    as.matrix() %>%
    apply(1, function(x) 100 * mean(x))
}

cellchange_results <- cellchange(
  satgpa,
  synth1$syn,
  c("sex", "sat_v")
)
# percents, uniques1, uniques2, matched  =
privacy_metric.cellchange(groundtruth, dataset, Qs, Xs)

length(cellchange_results$percents)

## [1] 4
```

There are 4 unique records in the original data that have an exact match in the synthetic data on the variables sex and sat_v, for a replicated uniques rate of about 0.4%.

```
mean(cellchange_results$percents)

## [1] 12.5
```

Among these 4 “apparent matches,” an average of 11.4% of other variables match between the two datasets.

```
max(cellchange_results$percents)

## [1] 50
```

The maximum match rate is 50%, or two out of the four other variables. Thus, these synthetic data have a low risk of perceived disclosure.

Tune the model and reevaluate

According to the synthpop vignette on utility measures, changing the order in which variables are synthesized and stratifying the synthesis are often the most useful tuning strategies, so we will try each of these approaches.

Change variable order

Let's move sat_m to the beginning of the visit sequence (which means that it will be directly sampled rather than synthesized).

```
synth2 <- syn(
  satgpa,
  minnumlevels = 2,
  method = c(
    sex = "cart", sat_v = "cart", sat_m = "sample",
    sat_sum = "~I(sat_v + sat_m)", hs_gpa = "cart", fy_gpa = "cart"
  ),
  visit.sequence = c("sat_m", "sex", "sat_v", "sat_sum", "hs_gpa", "fy_gpa")
)

##
## Variable(s): sex numeric but with only 2 or fewer distinct values turned
into factor(s) for synthesis.
##
## Synthesis
## -----------
##  sat_m sex sat_v sat_sum hs_gpa fy_gpa

utility2 <- utility.gen(synth2, satgpa)

## Running 50 permutations to get NULL utilities and printing every 10th.
## synthesis 1   10 20 30 40 50

utility2$pMSE

## [1] 0.06327193
```

This gives us a worse pMSE score. Which variables have problems?

```
compare(
  synth2,
  as.data.frame(satgpa),
  utility.stats = c("S_pMSE", "df"),
```

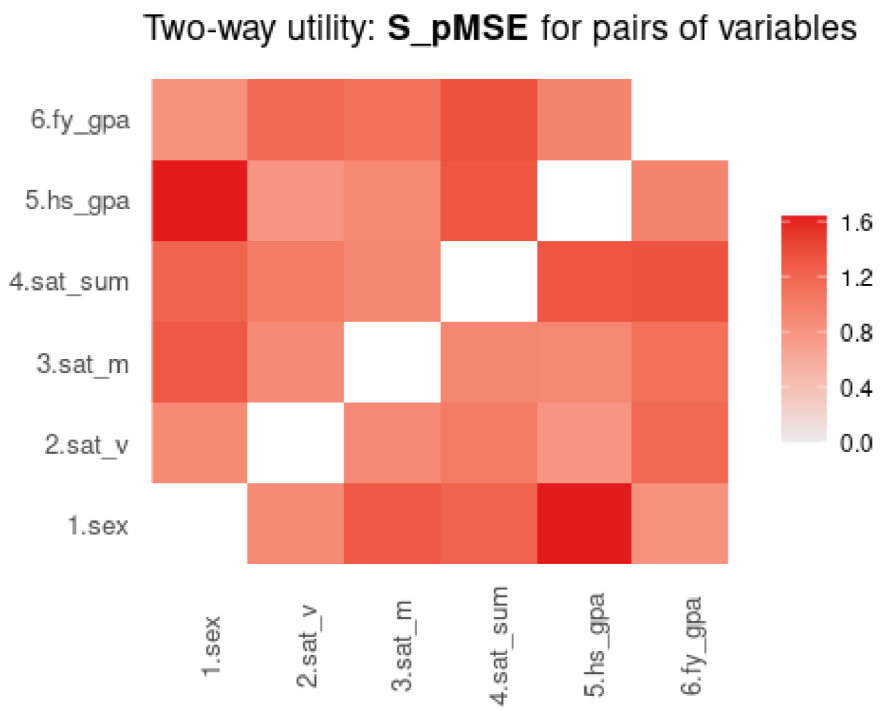
```
  plot = FALSE
)

##
## Comparing percentages observed with synthetic
##
##
## Selected utility measures:
##           S_pMSE df
## sex     0.144208  1
## sat_v   0.485188  4
## sat_m   1.034882  4
## sat_sum 0.494544  4
## hs_gpa  1.633315  4
## fy_gpa  0.372004  4
```

hs_gpa has the worst score now, but sat_m has improved at least. What about two-way tables?

```
twoway_tabs2 <- utility.tables(
  synth2,
  as.data.frame(satgpa),
  tables = "twoway"
)
twoway_tabs2

##
## Two-way utility: S_pMSE value plotted for 15 pairs of variables.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
##     1.sex:5.hs_gpa 4.sat_sum:6.fy_gpa 4.sat_sum:5.hs_gpa
1.sex:3.sat_m    1.sex:4.sat_sum
##             1.6398             1.3453             1.3245
1.3032             1.2153
```




```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians  Maxima
## pMSE    0.0013  0.0020
## S_pMSE  1.0234  1.6398
## df     24.0000 24.0000
##
## For more details of all scores use print.tabs = TRUE.
```

Sex by hs_gpa has the largest discrepancies. Perhaps we should try moving sex back to the beginning of the visit sequence, followed by sat_m:

```
synth3 <- syn(
  satgpa,
  minnumlevels = 2,
  method = c(
    sex = "sample", sat_v = "cart", sat_m = "cart",
    sat_sum = "~I(sat_v + sat_m)", hs_gpa = "cart", fy_gpa = "cart"
  ),
  visit.sequence = c("sex", "sat_m", "sat_v", "sat_sum", "hs_gpa", "fy_gpa")
)

##
## Variable(s): sex numeric but with only 2 or fewer distinct values turned
into factor(s) for synthesis.
##
```

```
## Synthesis
## -----------
##  sex sat_m sat_v sat_sum hs_gpa fy_gpa
```

```
utility3 <- utility.gen(synth3, satgpa)
```

```
## Running 50 permutations to get NULL utilities and printing every 10th.
## synthesis 1   10 20 30 40 50
```

```
utility3$pMSE
```

```
## [1] 0.04791638
```

This score indicates an improvement over synth2, but is no better than synth1. Does sat_m look improved over synth1 at least?

```
compare(
  synth3,
  as.data.frame(satgpa),
  utility.stats = c("S_pMSE", "df"),
  plot = FALSE
)
```

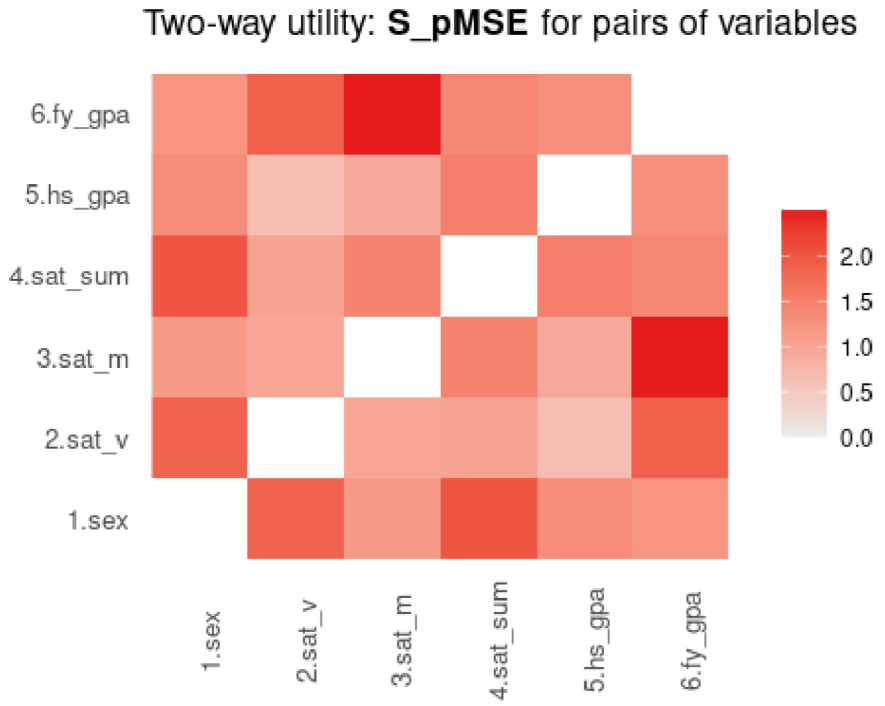
```
##
## Comparing percentages observed with synthetic
##
##
## Selected utility measures:
##           S_pMSE df
## sex     0.785662  1
## sat_v   1.183383  4
## sat_m   0.318683  4
## sat_sum 1.870847  4
## hs_gpa  0.637703  4
## fy_gpa  1.744717  4
```

Yes, sat_m has improved, but sat_sum and fy_gpa are noticeably worse. How does synth3 compare to synth1 in two-way and three-way tables?

```
twoway_tabs3 <- utility.tables(
  synth3,
  as.data.frame(satgpa),
  tables = "twoway"
)
twoway_tabs3
```

```
##
## Two-way utility: S_pMSE value plotted for 15 pairs of variables.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
```

```
##   3.sat_m:6.fy_gpa    1.sex:4.sat_sum   2.sat_v:6.fy_gpa
1.sex:2.sat_v 4.sat_sum:5.hs_gpa
##             2.4999             2.0158             1.8999
1.8495             1.5123
```

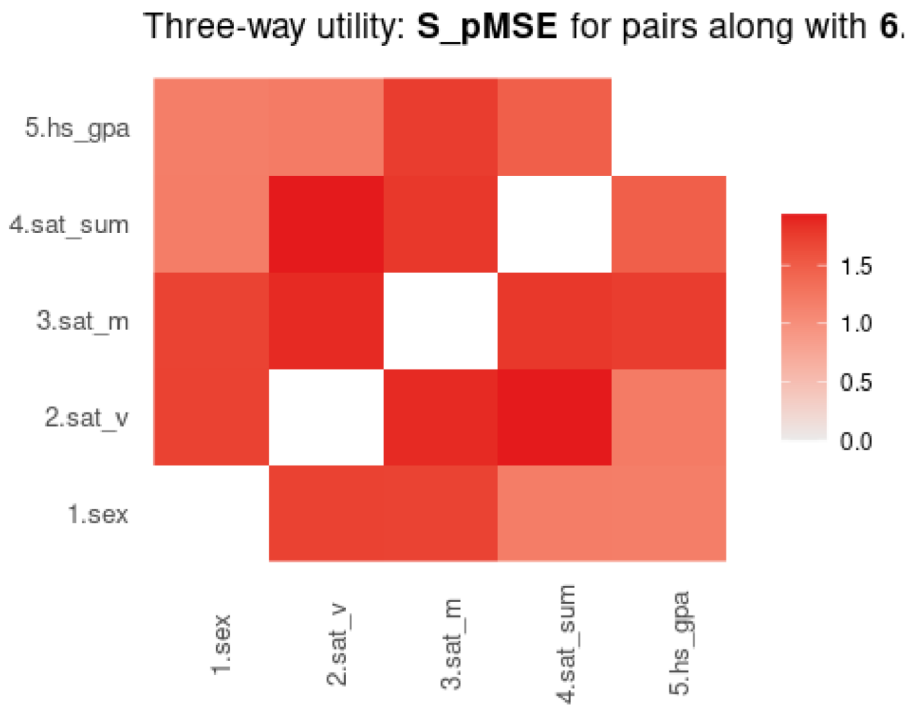



```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians  Maxima
## pMSE    0.0014  0.0037
## S_pMSE  1.3232  2.4999
## df     24.0000 24.0000
##
## For more details of all scores use print.tabs = TRUE.
```

The scores for two-way tables are noticeably worse than in synth1, including the median S_pMSE.

```
threeway_tabs3 <- utility.tables(
  synth3,
  as.data.frame(satgpa),
  tables = "threeway",
  third.var = "fy_gpa"
)
threeway_tabs3
```

```
##
## Three-way utility (total of 20 variable combinations):
##
## Average of 3-way scores S_pMSE (ordered) for 3-way tables including each
variable.
##  6.fy_gpa   3.sat_m 4.sat_sum   2.sat_v  5.hs_gpa     1.sex
##  1.585899  1.487992  1.474007  1.387737  1.342394  1.334060
##
## Variable with highest average score, 6.fy_gpa, selected to make plots.
## To see others, set parameter 'third.var'.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
## 2.sat_v:4.sat_sum:6.fy_gpa   2.sat_v:3.sat_m:6.fy_gpa
3.sat_m:4.sat_sum:6.fy_gpa  3.sat_m:5.hs_gpa:6.fy_gpa
##                     1.9359                     1.8691
1.7895                     1.7545
##     1.sex:2.sat_v:6.fy_gpa
##                     1.7208
```




```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians   Maxima
## pMSE    0.0053   0.0144
## S_pMSE  1.4334   1.9359
## df     49.0000 123.0000
```

```
##
## For more details of all scores use print.tabs = TRUE.
```

The three-way tables look worse overall as well. What about the three-way table for sat_sum by hs_gpa by fy_gpa specifically, since that seems like a likely variable-set of interest?

```
threeway_tabs3$tabs[row.names(threeway_tabs3$tabs) ==
"4.sat_sum:5.hs_gpa:6.fy_gpa", ]

##         pMSE      S_pMSE          df
##   0.0111006   1.4800797 120.0000000
```

At 1.48, the S_pMSE score for this table for synth3 is virtually the same as the corresponding score for synth1 (1.53).

In sum, none of our attempts to manipulate variable order improved the utility of the resulting synthetic data.

Stratify the model

In this dataset, it might make sense to stratify by sex.

```
synth4 <- syn.strata(
  satgpa,
  strata = "sex",
  minnumlevels = 2,
  method = c(
    sex = "sample", sat_v = "cart", sat_m = "cart",
    sat_sum = "~I(sat_v + sat_m)", hs_gpa = "cart", fy_gpa = "cart"
  )
)

## Number of observations in strata (original data):
##   1   2
## 516 484
##
## m = 1, strata = 1
## ------------------------------------------------------
## Sample(s) of size 517 will be generated from original data of size 516.
##
##
## Variable(s): sex numeric but with only 2 or fewer distinct values turned
into factor(s) for synthesis.
##
## Variable sex has only one value so its method has been changed to
"constant".
## Variable sex removed as predictor because only one value.
##
```

```
## Method "cart" is not valid for a variable without predictors (sat_v)
## Method has been changed to "sample"
##
## Synthesis
## -----------
##  sex sat_v sat_m sat_sum hs_gpa fy_gpa
##
## m = 1, strata = 2
## ------------------------------------------------------
## Sample(s) of size 483 will be generated from original data of size 484.
##
##
## Variable(s): sex numeric but with only 2 or fewer distinct values turned
into factor(s) for synthesis.
##
## Variable sex has only one value so its method has been changed to
"constant".
## Variable sex removed as predictor because only one value.
##
## Method "cart" is not valid for a variable without predictors (sat_v)
## Method has been changed to "sample"
##
## Synthesis
## -----------
##  sex sat_v sat_m sat_sum hs_gpa fy_gpa
```

```
utility4 <- utility.gen(synth4, satgpa)
```

```
## Running 50 permutations to get NULL utilities and printing every 10th.
## synthesis 1   10 20 30 40 50
```

```
utility4$pMSE
```

```
## [1] 0.05519151
```

Stratifying by sex yields a worse pMSE score overall. How do the one-way, two-way, and three-way tables compare?

```
compare(
  synth4,
  as.data.frame(satgpa),
  utility.stats = c("S_pMSE", "df"),
  plot = FALSE
)
```

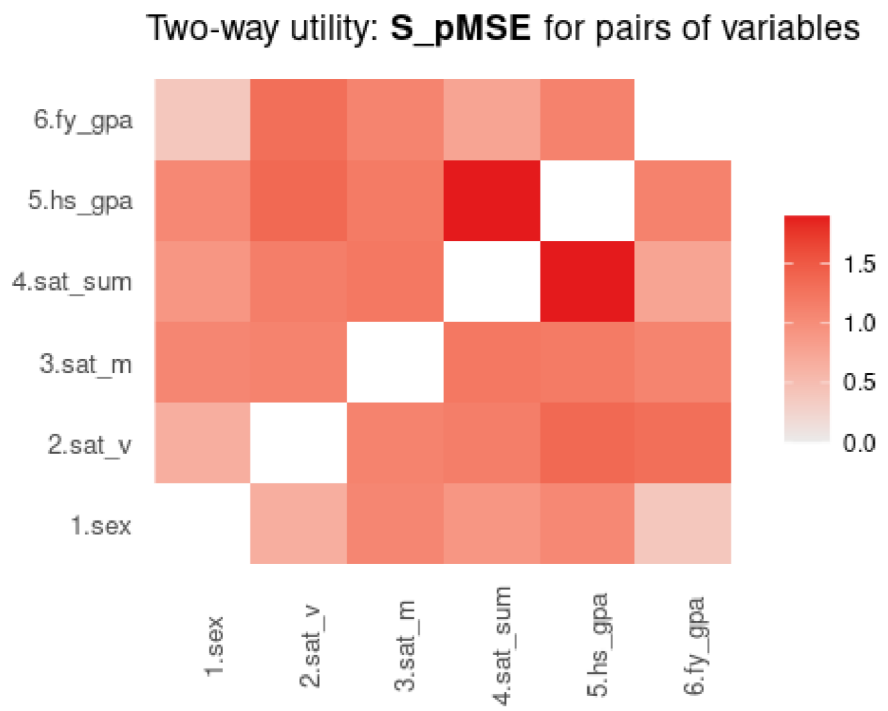
```
##
## Comparing percentages observed with synthetic
##
##
```

```
## Selected utility measures:
##           S_pMSE df
## sex     0.004004  1
## sat_v   1.107372  4
## sat_m   1.810368  4
## sat_sum 0.689388  4
## hs_gpa  0.475283  4
## fy_gpa  0.286088  4
```

sat_m still shows the greatest discrepancies.

```
twoway_tabs4 <- utility.tables(
  synth4,
  as.data.frame(satgpa),
  tables = "twoway"
)
twoway_tabs4
```

```
##
## Two-way utility: S_pMSE value plotted for 15 pairs of variables.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
## 4.sat_sum:5.hs_gpa   2.sat_v:5.hs_gpa   2.sat_v:6.fy_gpa
3.sat_m:4.sat_sum   3.sat_m:5.hs_gpa
##             1.9044             1.3681             1.3123
1.2258             1.1946
```



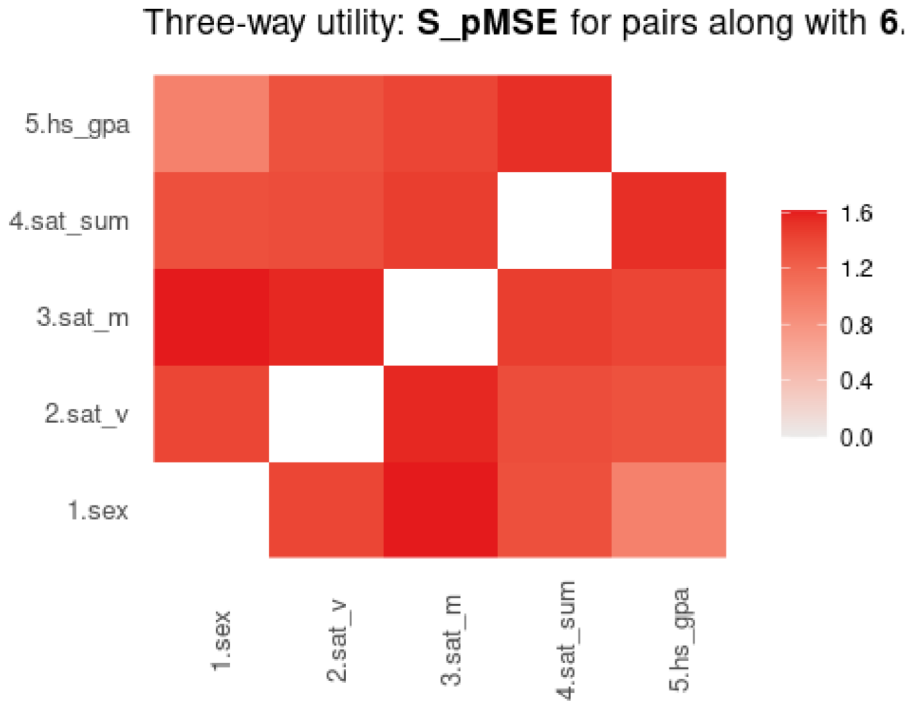

```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians  Maxima
## pMSE    0.0017  0.0029
## S_pMSE  1.1086  1.9044
## df     24.0000 24.0000
##
## For more details of all scores use print.tabs = TRUE.
```

We see a particularly poor match for the table of hs_gpa by sat_sum, which is an important relationship to preserve.

```
threeway_tabs4 <- utility.tables(
  synth1,
  as.data.frame(satgpa),
  tables = "threeway",
  third.var = "fy_gpa"
)
threeway_tabs4
```

```
##
## Three-way utility (total of 20 variable combinations):
##
## Average of 3-way scores S_pMSE (ordered) for 3-way tables including each
variable.
##  6.fy_gpa   3.sat_m 4.sat_sum  5.hs_gpa     1.sex   2.sat_v
```

```
##  1.397452  1.278826  1.195979  1.183172  1.177466  1.169369
##
## Variable with highest average score, 6.fy_gpa, selected to make plots.
## To see others, set parameter 'third.var'.
##
## Variable combinations with worst 5 utility scores (S_pMSE):
##      1.sex:3.sat_m:6.fy_gpa    2.sat_v:3.sat_m:6.fy_gpa
4.sat_sum:5.hs_gpa:6.fy_gpa  3.sat_m:4.sat_sum:6.fy_gpa
##                      1.6133                      1.5650
1.5324                      1.4553
##   3.sat_m:5.hs_gpa:6.fy_gpa
##                      1.4122
```




```
##
## Medians and maxima of selected utility measures for all tables compared
##        Medians   Maxima
## pMSE    0.0046   0.0118
## S_pMSE  1.2423   1.6133
## df     50.5000 124.0000
##
## For more details of all scores use print.tabs = TRUE.
```

Here, the median error is substantially higher than in the initial, untuned model, though the maximum error is lower.

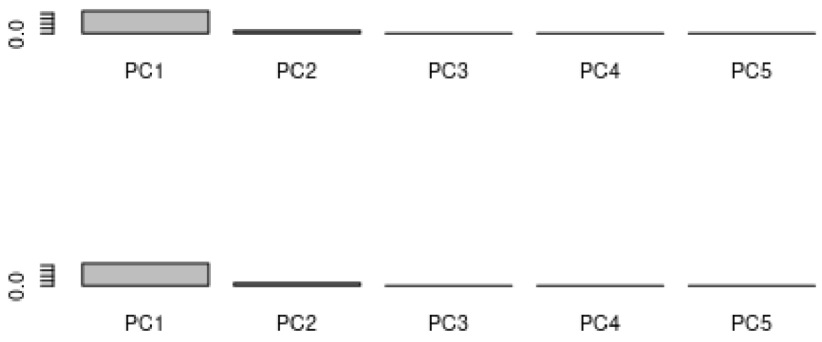
In sum, the stratified model does not appear to greatly improve the utility of the resulting synthetic dataset.

Analysis example: PCA

A principal components analysis is a common dimension reduction technique used in many types of analysis. How do the results of a PCA compare between the original and synthetic data?

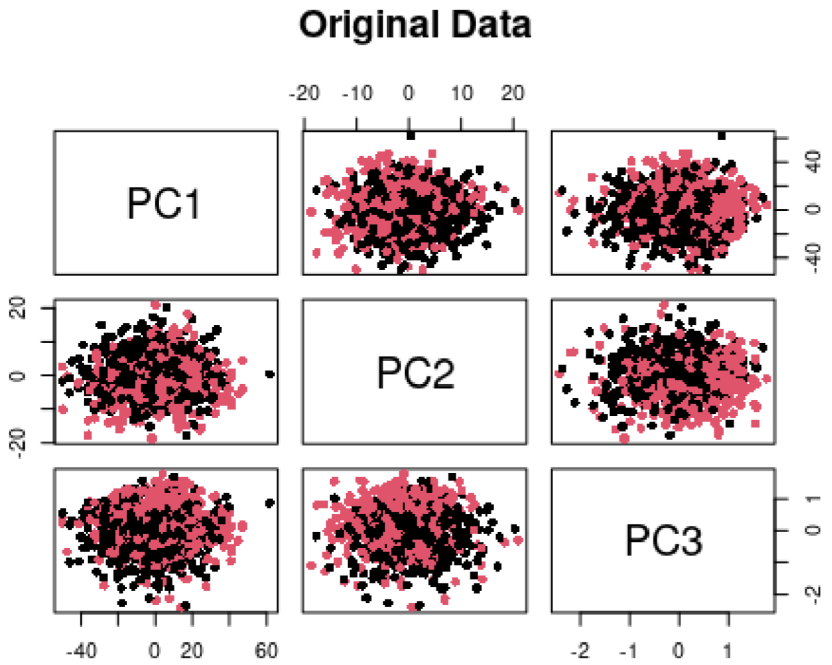
```
pca_orig <- prcomp(satgpa[ , 2:6])
pca_synth <- prcomp(synth1$syn[ , 2:6])

layout(matrix(1:3, ncol = 1))
barplot(summary(pca_orig)$importance[2,1:5])
barplot(summary(pca_synth)$importance[2,1:5])
```

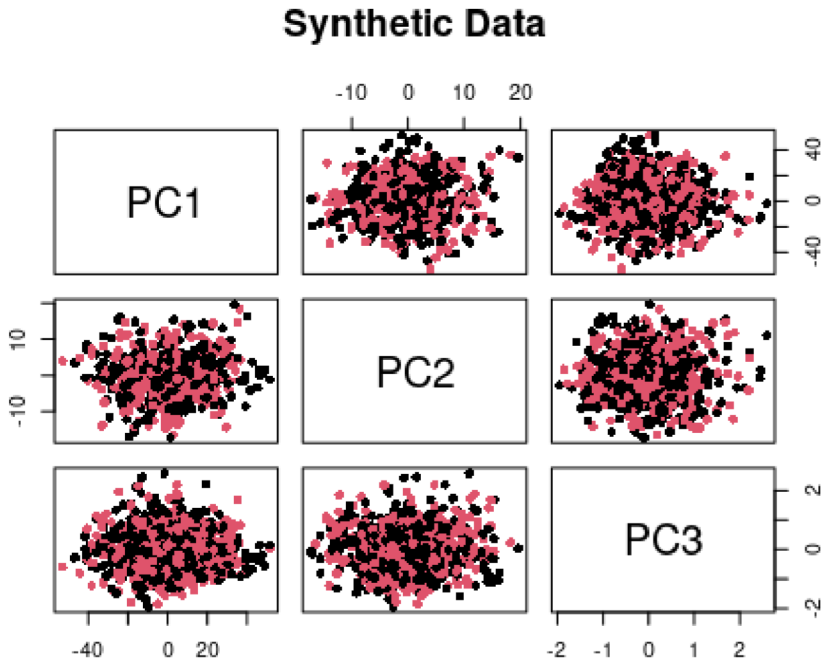


The barplot of the importance of components shows that the first three components capture the largest share of variance in the data. Let's use pair plots to visualize the relationships between these three components in the original and synthetic data, with differences by sex highlighted in the plotting point color.

```
pairs(pca_orig$x[,1:3], pch = 16, col = satgpa$sex, main = "Original Data")
```




```r
pairs(pca_synth$x[,1:3], pch = 16, col = satgpa$sex, main = "Synthetic Data")
```



These plots show that the synthetic data does an adequate job of mirroring the results of the PCA in the original data.

Conclusions

In sum, the fully conditional specification approach implemented by the synthpop package produces a synthetic version of the SAT GPA data with relatively high utility, even without substantial tuning. Moreover, there is a low rate of matching on the characteristics of apparent unique matches between the original and synthetic data, which translates into a low risk of perceived disclosure. Because of the high utility and low disclosure risk, this synthetic dataset would be well-suited to a variety of use cases, including public release, software testing, analysis testing, and educational use.