Inference from fitted models in **synthpop**

Gillian M Raab & Beata Nowok

Abstract

This paper describes the methods used for inference from models fitted to synthetic data generated by the **synthpop** package. Since earlier versions of **synthpop** new methodology for making inferences from synthetic data has been published [5]. These new methods have the advantage over previously proposed methods [6] of usually requiring only a single synthetic data set to be produced. We explain how these methods have been implemented in **synthpop** functions. Inference from synthetic data is only valid if the model used for synthesis has reproduced the relationships between variables that influence the fit to the model. Thus it is important that the results from synthetic data are compared to those from the original data. These methods, including some new, recently developed tests, are fully explained here.

1 Introduction

The **synthpop** package for creating synthetic data allows the user to produce synthetic version(s) of confidential data and also provides functions (e.g. `glm.synds()`) to make inferences from statistical models fitted to the synthetic data and to compare the results with those from a fit to the original data (e.g. `compare.fit.synds()`). This vignette explains the statistics computed by these functions.

The main focus of the **synthpop** package [4] is to produce data for exploratory analysis with results for publication produced when the code developed on the synthetic data is run on the original, confidential data. This use of **synthpop** will usually require only a single synthetic data set (`m = 1`) with the same number of records (`k`) as the original (`n`), the default setting in **synthpop**. For a user who simply wants to estimate the reults that would be obtained from the original data, if they were available,an analysis of the synthetic data set, as if it were the original, is all that

is required. No special methods of estimation are needed for the parameters of models fitted to synthetic data. When `m > 1` the results from the different syntheses need to be combined and when the size of the synthetic data differs from the real data (`k` $\neq$ `n`) the standard errors of the coefficients from the synthetic fit require adjustment to estimate the results that would be obtained from the original data. The methods implemented in **synthpop** for these situations are described in Section 2.

The other situation is when the user wishes to make inferences to the population parameters directly from the synthetic data. For this case the standard errors of the estimates must include the contributions to the uncertainty of the estimates from two sources:

1. The difference between the parameters estimated from the synthetic data and from the original data;

2. The difference between the estimates from the original data and the parameters of the model assumed to have generated the synthetic data.

The contribution from the first source decreases as the number of synthetic data sets increases, while that from the second source is unchanged. The second corresponds to the estimated standard error that would have been obtained from fitting the model to the original data. Methods for population inference are dependent on the details of how the synthetic data have been produced, as discussed in [1] and recently developed in [5]. The functions `glm.synds()`, `lm.synds()`, `polr.synds()` and `multinom.synds()` in **synthpop**, and their summary function `summary.fit.synds()` implement these methods . If synthesis has used samples from the posterior predictive distribution of the population, given the observed data (parameter `proper` of the function `syn()`) this will be recognized by `summary.fit.synds()` and appropriate calculations performed. If the model being fitted includes some variables that have not been synthesised and have not been used in synthesising models for all other variables (not in the visit.sequence or not at the start of the visit.sequence with `method == ""`), the fitting function will return a component `$incomplete` as TRUE. This causes the functions `summary.fit.synds()` and `compare.fit.synds()`) used on the fit to use methods appropriate to incomplete/partially synthetic data [7]. Details of how to carry out population inference with **synthpop** functions are in Section 3.

To emphasize the difference between inference to the expectation from the real data and inference to the population quantities the estimates and their standard errors are labelled differently

in **synthpop** inference, as follows

Inference for	Coefficient	Standard error	z statistic
Coefficients expected from original data	`xpct(Beta)`	`xpct(se.Beta)`	`xpct(z)`
Population coefficients	`Beta.syn`	`se.Beta.syn`	`z.syn`

The final Section 4 explains the statistics calculated when the results from a fit of a model to synthetic data are compared to those from original with the function `compare.fit.synds()`. The synthetic data estimates may differ from what will be obtained from the real data if the model used for the synthesis has not captured all of the relationships between variables that influence the fitted parameters. The function `compare.fit.synds()` requires access to the original data as well as the synthetic data. It is designed to be used by the staff producing the synthetic data, or by the user at a validation step, with the long-term goal of evaluating and improving synthesis methods.

The following notation is used for quantities used in the calculations explained below.

- Q - the "true" vector of coefficients in the population from which the original data are assumed to be a sample.
- p - the number of coefficients and length of the vector Q.
- $q_1, \cdots, q_i, \cdots, q_m$ - vectors of estimated coefficients, where q_i is the estimate from the fit to the i^{th} synthesis and their mean vector is $\bar{q}_m = \sum_{i=1}^{m} q_i/m$.
- $\hat{V}_{orig}$ - the estimated variance-covariance matrix of the parameter estimates from a fit of the model to the original data; its diagonal vector is denoted by v_{orig}.
- $\hat{V}_1, \cdots, \hat{V}_i, \cdots, \hat{V}_m$ - the estimated variance-covariance matrices of the coefficients q_i, calculated from each synthetic data set as if it were the original, and their mean matrix $\bar{V}_m = \sum_{i=1}^{m} \hat{V}_i/m$.
- $v_1, \cdots, v_i, \cdots, v_m$ - the diagonal vectors of $\hat{V}_i$ which give the estimated variances of the coefficients q_i and their mean vector $\bar{v}_m = \sum_{i=1}^{m} v_i/m$.
- $b_m = \sum_{i=1}^{m} (q_i - \bar{q}_m)^2/(m-1)$ - the between-synthesis variance of the q_i.
- $B_m = b'_{mat} b_{mat}/(m-1)$, where b_{mat} is a matrix of differences of the coefficients from their mean with i^{th} row $b_{mat}[i,\,] = q_i - \bar{q}_m$. The diagonal of B_m is b_m.

In all the situations discussed here Q is estimated by $\bar{q}_m$ but its variance and standard errors will depend on what type of inference is required and how the synthesis has been carried out.

The results here apply when both the original and synthetic data are analysed by methods appropriate for simple random sampling. They could also be used when both analyses use the same methods for complex samples, as discussed in [5], but methods for complex samples are not currently implemented in **synthpop**.

2 Inference to results from the original data

Here $\bar{q}_m$ and $\sqrt{\bar{v}_m}$ estimate $\hat{Q}$ and $\sqrt{v_{orig}}$, the coefficients and their standard errors from a fit to the original data. For a single synthetic data set produced by `syn()` with `m = 1` and `k = n` and the fit summarised with the default setting `population.inference = FALSE` there is no need to use the special functions for inference from synthetic data. These are the default settings in `syn()` and in `summary.fit.synds()`, used to carry out inference from an object produced by `glm.synds()` or `lm.synds()`.

```
R> library(synthpop)
R> ods <- SD2011[, c("smoke", "sex", "age", "edu")]
R> levels(ods$edu) <- c("NONE", "VOC", "SEC", "HIGH")
R> s1 <- syn(ods, seed = 1234)

Synthesis
-----------
 smoke sex age edu

R> summary(glm(smoke ~ sex + age + edu + sex * edu,
+   data = s1$syn, family = "binomial"))

Call:
glm(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s1$syn)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.974  -1.329   0.634   0.814   1.054

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       0.23440    0.15712    1.49   0.1357
sexFEMALE         0.87392    0.14809    5.90  3.6e-09 ***
age               0.00488    0.00195    2.51   0.0121 *
```

```
eduVOC            -0.01597    0.13040   -0.12   0.9025
eduSEC             0.47012    0.14029    3.35   0.0008 ***
eduHIGH            0.96509    0.17315    5.57  2.5e-08 ***
sexFEMALE:eduVOC  -0.34922    0.18419   -1.90   0.0580 .
sexFEMALE:eduSEC  -0.19880    0.19472   -1.02   0.3073
sexFEMALE:eduHIGH -0.66688    0.22817   -2.92   0.0035 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5693.7  on 4976  degrees of freedom
Residual deviance: 5495.3  on 4968  degrees of freedom
  (23 observations deleted due to missingness)
AIC: 5513

Number of Fisher Scoring iterations: 4

R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s1, family = "binomial"))

Fit to synthetic data set with a single synthesis. Inference to coefficients
and standard errors that would be obtained from the original data.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s1)

Combined estimates:
                  xpct(Beta) xpct(se.Beta) xpct(z) Pr(>|xpct(z)|)
(Intercept)          0.23440       0.15712    1.49         0.1357
sexFEMALE            0.87392       0.14809    5.90        3.6e-09 ***
age                  0.00488       0.00195    2.51         0.0121 *
eduVOC              -0.01597       0.13040   -0.12         0.9025
eduSEC               0.47012       0.14029    3.35         0.0008 ***
eduHIGH              0.96509       0.17315    5.57        2.5e-08 ***
sexFEMALE:eduVOC    -0.34922       0.18419   -1.90         0.0580 .
sexFEMALE:eduSEC    -0.19880       0.19472   -1.02         0.3073
sexFEMALE:eduHIGH   -0.66688       0.22817   -2.92         0.0035 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Running this code shows that the results from `glm()` and `glm.synds()` are identical. This is also the case if the synthesis is carried out using the parameter `proper = TRUE` of `syn()` or if some of the variables in the model have not been synthesised. One small advantage of using `glm.synds()`

is that it checks and gives warnings if the model includes unsynthesised variables that are not at the start of the visit sequence, which can produce results which are wrong, as in the next example.

```
R> s2 <- syn(ods, seed = 1234,visit.sequence=c("smoke","edu","age"))
Variable(s): sex not synthesised or used in prediction.
CAUTION: The synthesised data will contain the variable(s) unchanged.

Synthesis
-----------
 smoke edu age
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+ data = s2, family = "binomial"))
************************************************************
WARNING: Some variable(s) in formula (model to be fitted)
are not synthesised and not used in synthesising models
for all other variables: sex
Methods in synthesis order are:
   smoke      edu      age      sex
"sample"   "cart"   "cart"       ""
Results may not be correct.
************************************************************
Fit to synthetic data set with a single synthesis. Inference to coefficients
and standard errors that would be obtained from the original data.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s2)

Combined estimates:
                  xpct(Beta) xpct(se.Beta) xpct(z) Pr(>|xpct(z)|)
(Intercept)          0.79600       0.16398    4.85        1.2e-06 ***
sexFEMALE            0.01161       0.14730    0.08          0.937
age                  0.00352       0.00193    1.82          0.068 .
eduVOC              -0.22653       0.14289   -1.59          0.113
eduSEC               0.13920       0.14781    0.94          0.346
eduHIGH              0.87571       0.18405    4.76        2.0e-06 ***
sexFEMALE:eduVOC     0.00139       0.18238    0.01          0.994
sexFEMALE:eduSEC    -0.01626       0.19114   -0.09          0.932
sexFEMALE:eduHIGH   -0.22723       0.23468   -0.97          0.333
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The variable `sex` has been omitted from the `visit.sequence`, so that its relationships to other variables are not maintained in the synthetic data. Warnings appear and the results show that the coefficients involving `sex` are no longer significant and the other coefficients have changed.

The only time when `glm.synds()` will give different results from `glm()` with `population.inference = FALSE` is when `k` and `n` differ. In this case the standard errors of the coefficients are adjusted to what would be expected from the original data.

```
R> s3 <- syn(ods, seed = 1234, k = 500)

Sample(s) of size 500 will be generated from original data of size 5000.

Synthesis
-----------
 smoke sex age edu

R> ## analysing synthetic data with just glm()
R> summary(glm(smoke ~ sex + age + edu + sex * edu,
+              data = s3$syn, family = "binomial"))

Call:
glm(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s3$syn)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.020  -1.143   0.611   0.811   1.292

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.36424    0.48105   -0.76  0.44894
sexFEMALE          1.52487    0.45421    3.36  0.00079 ***
age                0.00611    0.00618    0.99  0.32278
eduVOC             0.75458    0.39252    1.92  0.05455 .
eduSEC             0.93355    0.42495    2.20  0.02803 *
eduHIGH            1.78241    0.58580    3.04  0.00234 **
sexFEMALE:eduVOC  -1.17474    0.57537   -2.04  0.04118 *
sexFEMALE:eduSEC  -0.64500    0.60501   -1.07  0.28638
sexFEMALE:eduHIGH -2.12138    0.74579   -2.84  0.00445 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 573.92  on 497  degrees of freedom
Residual deviance: 547.63  on 489  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 565.6

Number of Fisher Scoring iterations: 4
```

```
R> ## using glm.synds()
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+                    data = s3, family = "binomial"))

Fit to synthetic data set with a single synthesis. Inference to coefficients
and standard errors that would be obtained from the original data.

The synthetic data have a different size (500) from the original data (5000),
so the standard errors of the coefficients have been adjusted to estimate
the standard errors from the original data.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s3)

Combined estimates:
                  xpct(Beta) xpct(se.Beta) xpct(z) Pr(>|xpct(z)|)
(Intercept)         -0.36424       0.15212   -2.39        0.01665 *
sexFEMALE            1.52487       0.14363   10.62        < 2e-16 ***
age                  0.00611       0.00195    3.13        0.00177 **
eduVOC               0.75458       0.12412    6.08        1.2e-09 ***
eduSEC               0.93355       0.13438    6.95        3.7e-12 ***
eduHIGH              1.78241       0.18525    9.62        < 2e-16 ***
sexFEMALE:eduVOC    -1.17474       0.18195   -6.46        1.1e-10 ***
sexFEMALE:eduSEC    -0.64500       0.19132   -3.37        0.00075 ***
sexFEMALE:eduHIGH   -2.12138       0.23584   -9.00        < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here `glm()` fitted to a synthetic sample of size `k = 500` that is smaller than the original `n = 5000` gives larger standard errors, with only a few coefficients appearing significant, whereas `glm.synds()` estimates what would be found from the original larger sample. If `m > 1` then using `glm.synds()` will give you the average of these results for all `m` syntheses, as well as results from selected individual syntheses if you specify the parameter `msel`. Results from individual syntheses are simply what the standard functions `lm()` or `glm()` would report.

3 Inference to population parameters

This code gets inference to population parameters for our example with all variables synthesised.

```
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s1, family = "binomial"), population.inference = TRUE)
```

```
Fit to synthetic data set with a single synthesis. Inference to population
coefficients when all variables in the model are synthesised.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s1)

Combined estimates:
                  Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)        0.23440     0.22220  1.05        0.291
sexFEMALE          0.87392     0.20944  4.17      3.0e-05 ***
age                0.00488     0.00275  1.77        0.076 .
eduVOC            -0.01597     0.18441 -0.09        0.931
eduSEC             0.47012     0.19840  2.37        0.018 *
eduHIGH            0.96509     0.24487  3.94      8.1e-05 ***
sexFEMALE:eduVOC  -0.34922     0.26049 -1.34        0.180
sexFEMALE:eduSEC  -0.19880     0.27537 -0.72        0.470
sexFEMALE:eduHIGH -0.66688     0.32268 -2.07        0.039 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You will find that, as expected, the standard errors are larger than when `population.inference = FALSE`. When, as for `s1`, synthesis has been carried out with the default `proper = FALSE` the standard errors of the average estimated coefficients, $\bar{q}_m$ are calculated as $\sqrt{\bar{v}_m/m + \bar{v}_m k/n)}$. If the data had been synthesised with the `syn()` option `proper = TRUE` this will be recognized by `glm.fit.synds()` and its `summary()` function and appropriate standard errors for $\bar{q}_m$ calculated as $\sqrt{\bar{v}_m(1 + k/n)/m + \bar{v}_m k/n}$. In each case the first term under the square root sign gives the contribution from the differences between $\bar{q}_m$ and $\hat{Q}$ while the second provides the contribution from the differences between $\hat{Q}$ and the population parameters Q. The first term is larger when synthesis is done with `proper = TRUE` as we see from the next analysis.

```
R> s4 <- syn(ods, seed = 5678, proper = TRUE)

Synthesis
-----------
 smoke sex age edu

R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s4, family = "binomial"), population.inference = TRUE)

Fit to synthetic data set with a single synthesis. Inference to population
coefficients when all variables in the model are synthesised.
```

```
Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s4)

Combined estimates:
                  Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)       -0.09316     0.26848 -0.35      0.72859
sexFEMALE          0.92148     0.26242  3.51      0.00045 ***
age                0.00935     0.00341  2.75      0.00604 **
eduVOC             0.26584     0.22889  1.16      0.24546
eduSEC             0.56847     0.24238  2.35      0.01901 *
eduHIGH            1.30446     0.31426  4.15      3.3e-05 ***
sexFEMALE:eduVOC  -0.48374     0.32551 -1.49      0.13725
sexFEMALE:eduSEC  -0.31451     0.33729 -0.93      0.35109
sexFEMALE:eduHIGH -0.79185     0.41273 -1.92      0.05504 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see here that inference is possible when only one synthetic data set is produced, but increasing `m` improves the precision of the estimates as you will see from the smaller `se(Beta.syn)` values in the example below.

```
R> s5 <- syn(ods, seed = 5678, m = 10, proper = TRUE, print.flag = FALSE)
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s5, family = "binomial"), population.inference = TRUE)

Fit to synthetic data set with 10 syntheses. Inference to population
coefficients when all variables in the model are synthesised.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s5)

Combined estimates:
                  Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)        0.12996     0.17508  0.74       0.4579
sexFEMALE          1.02161     0.16885  6.05      1.4e-09 ***
age                0.00530     0.00213  2.48       0.0130 *
eduVOC             0.14992     0.14708  1.02       0.3080
eduSEC             0.48663     0.15828  3.07       0.0021 **
eduHIGH            0.98525     0.19689  5.00      5.6e-07 ***
sexFEMALE:eduVOC  -0.51278     0.20700 -2.48       0.0132 *
sexFEMALE:eduSEC  -0.41484     0.21636 -1.92       0.0552 .
sexFEMALE:eduHIGH -0.63809     0.25982 -2.46       0.0141 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The **synthpop** package was written with completely synthesised data in mind but, with care, it can also be used for population inference for some incompletely synthesised data, when all values for one or more variable are unchanged. The synthesis must have been conditional on the values of all unsynthesised variables. This can be done by placing them at the start of the visit sequence and setting their parameter `method = ""`. The functions `lm.synds()` and `glm.synds()` check this condition and print warnings, as we saw for the synthesis `s2` above.

When some of the variables in the formula are unchanged the between synthesis variance can be estimated from multiple syntheses [7] with standard errors calculated as $\sqrt{b_m/m + \bar{v}_m k/n}$. This standard error estimate requires $m > 1$ and larger values of m such as $m > 5$ are recommended for reliable results. In the next example only one of the variables in the formula has been synthesised.

```
R> s6 <- syn(ods, seed = 910011, m = 12,
+   method = c("", "", "", "cart"), print.flag = FALSE)
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s6, family = "binomial"), population.inference = TRUE)

Fit to synthetic data set with 12 syntheses. Inference to population
coefficients when all variables in the model are synthesised.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s6)

Combined estimates:
                  Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)        0.08885     0.16709  0.53      0.59491
sexFEMALE          1.05974     0.15927  6.65      2.9e-11 ***
age                0.00582     0.00204  2.86      0.00429 **
eduVOC             0.16052     0.13947  1.15      0.24977
eduSEC             0.53571     0.15147  3.54      0.00041 ***
eduHIGH            1.15776     0.19147  6.05      1.5e-09 ***
sexFEMALE:eduVOC  -0.56293     0.19543 -2.88      0.00397 **
sexFEMALE:eduSEC  -0.44036     0.20591 -2.14      0.03247 *
sexFEMALE:eduHIGH -0.94733     0.24693 -3.84      0.00012 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R>
```

If a user attempts an incomplete fit with `population.inference = TRUE` from a synthesis with `m = 1`, the program will detect this and revert to inference from a complete synthesis with a resulting increase in the standard errors as we show here.

```
R> s7 <- syn(ods, seed = 910011, m = 1,
+   method = c("", "", "", "cart"), print.flag = FALSE)
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s7, family = "binomial"), population.inference = TRUE)

Fit to synthetic data set with a single synthesis. Inference to population
coefficients when all variables in the model are synthesised.


Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s7)


Combined estimates:
                 Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)       0.01936     0.22223  0.09      0.93057
sexFEMALE         1.26646     0.21804  5.81      6.3e-09 ***
age               0.00559     0.00276  2.03      0.04272 *
eduVOC            0.33485     0.18656  1.79      0.07267 .
eduSEC            0.50242     0.19975  2.52      0.01190 *
eduHIGH           1.25892     0.25646  4.91      9.2e-07 ***
sexFEMALE:eduVOC -0.95996     0.26793 -3.58      0.00034 ***
sexFEMALE:eduSEC -0.58567     0.27722 -2.11      0.03463 *
sexFEMALE:eduHIGH -1.10564     0.33767 -3.27      0.00106 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R>
```

In addition to these combined estimates from all the syntheses, the user can choose to print out the results from selected individual syntheses from the `m` with the parameter `(msel)`. These are simply what the original functions (`lm()` or `glm()`) would report.

```
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s6, family = "binomial"), population.inference = TRUE,
+   incomplete = TRUE, msel = 1:2)
```

```
Fit to synthetic data set with 12 syntheses. Inference to population
coefficients when all variables in the model are synthesised.

Call:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s6)

Combined estimates:
                  Beta.syn se.Beta.syn z.syn Pr(>|z.syn|)
(Intercept)        0.08885     0.16709  0.53      0.59491
sexFEMALE          1.05974     0.15927  6.65      2.9e-11 ***
age                0.00582     0.00204  2.86      0.00429 **
eduVOC             0.16052     0.13947  1.15      0.24977
eduSEC             0.53571     0.15147  3.54      0.00041 ***
eduHIGH            1.15776     0.19147  6.05      1.5e-09 ***
sexFEMALE:eduVOC  -0.56293     0.19543 -2.88      0.00397 **
sexFEMALE:eduSEC  -0.44036     0.20591 -2.14      0.03247 *
sexFEMALE:eduHIGH -0.94733     0.24693 -3.84      0.00012 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimates for selected syntheses contributing to the combined estimates:

Coefficients:
                     syn=1     syn=2
(Intercept)        0.019362  0.062836
sexFEMALE          1.266456  1.008554
age                0.005587  0.006237
eduVOC             0.334849  0.189847
eduSEC             0.502420  0.501606
eduHIGH            1.258915  1.085128
sexFEMALE:eduVOC  -0.959960 -0.645905
sexFEMALE:eduSEC  -0.585671 -0.222160
sexFEMALE:eduHIGH -1.105639 -0.773811

z values:
                    syn=1   syn=2
(Intercept)        0.1232  0.4002
sexFEMALE          8.2142  6.7988
age                2.8658  3.1928
eduVOC             2.5384  1.4429
eduSEC             3.5570  3.4912
eduHIGH            6.9420  6.3346
sexFEMALE:eduVOC  -5.0669 -3.5127
sexFEMALE:eduSEC  -2.9878 -1.1305
sexFEMALE:eduHIGH -4.6306 -3.4075
```

4 Comparing fits to the original and synthesised data

4.1 Overview of comparisons

The inferences from synthetic data, described above, all depend on the synthesising model being correct. If the original data can be accessed then the results of a model fitted to the original and synthetic data can be compared with the function `compare.fit.synds()` which provides a graphical representation of the differences as well as statistics to evaluate their importance. The methods we implement for evaluating these biases are standardised differences in the coefficients and a combined lack-of-fit test for the differences between the vector of coefficients estimated from the original and the synthetic data. These can be used to evaluate the specific utility of synthetic data for this analysis and they incorporate statistical tests of the null hypothesis that the synthesising model is the same as the model that is presumed to have generated the original data. Section 4.2 explains the details of these methods.

The graphical output from `compare.fit.synds()` consists of a comparison of the confidence intervals for the coefficients calculated from the original and the synthetic data. Results also include a confidence interval overlap measure for each coefficient: the ratio of the overlap of the intervals to an average of their lengths. Previous use of confidence-interval-overlap measures [1, 2, 8] have compared the intervals for population inference with synthetic data to the interval from the original data. The parameter `population.inference = TRUE` can be used to plot and calculate these intervals. But in our default case with `population.inference = FALSE` plots and intervals for the estimate of $\hat{Q}$ expected from synthetic data are calculated. Details of each method are explained in Sections 4.3 and 4.4.

4.2 Assessing the bias in estimates from synthetic data

Function *compare.fit.synds*() calculates the standardised difference between the original and synthetic coefficients where the standardisation uses the standard errors of the original fit, $\sqrt{v_{orig}}$. In Section 2 it was necessary to calculate the standard errors from an estimate $\bar{V}_m$ of $\hat{V}_{orig}$, but here we have access to the original data and so we can use its known value. The vector of the means of the standardised differences, z_j, over m synthetic data sets is $z = (\bar{q}_m - \hat{Q})/\sqrt{v_{orig}/m}$. The mean of the absolute values of z_j for each coefficient, the $|\bar{z}| = \Sigma_{j=1}^{p}|z_j|/p$ value, gives the first summary measure of the difference between the observed and synthetic data.

The standardised differences simply summarise the bias in the estimates in relation to the precision of the estimates from the original data. If the model used to synthesise the data was the one which generated the original data, for completely synthesised data with `proper = FALSE` the expected value of each $|z_j|$ and of $|\bar{z}|$ will be $\sqrt{2/(m\pi)}$[1] taking the values 0.798, 0.356, 0.252, 0.056 for $m = 1, 5, 10, 200$. For completely synthesised data with `proper = TRUE` the expectations of $|z_j|$ and of $|\bar{z}|$ become $\sqrt{(1 + 2/\pi)/m}$, giving larger absolute differences. As m increases the bias expected in the null case when the synthesising model is correct decreases and the test described below will have greater power to detect a poor synthesising model. If the synthesising model is correct the coefficients from the synthetic data $\bar{q}_m$ will have expectations $\hat{Q}$ and variances $v_{orig}(k/n)/m$ when the synthesis parameter `proper = FALSE` or $v_{orig}(1+k/n)/m$ for synthesis with `proper = TRUE` [5]. These results allow a test of the bias of each coefficient to be calculated as a p-value giving the probability of a difference as large as that observed under the null hypothesis that the synthesising model is correct.

A composite lack-of-fit test for the vector of differences between the coefficients from the fit to the original and synthetic data can be obtained from the quadratic form

$$LOF = (\bar{q}_m - \hat{Q})' {V_{diff}}^{-1} (\bar{q}_m - \hat{Q})$$

where V_{diff} is the between synthesis variance-covariance matrix for the differences, which depends on the synthesis method used. For completely synthesised data $V_{diff} = \hat{V}_{orig}(k/n)/m$ if the synthesis parameter `proper = FALSE` or $V_{diff} = \hat{V}_{orig}(1 + k/n)/m$ for synthesis with `proper = TRUE` [5]. If the synthesis model is correct these LOF measures will have a χ^2 distribution with degrees of freedom equal to the number of coefficients in the model.

We illustrate these results with three different syntheses and three different logistic regression models. For the first synthesis `s7` and the first fit `f7` the synthesis model is compatible with the model fitted. By using the new function `syn.strata` the synthesis preserves all the relationships between the stratifying variable, synthesised first, and the within-stratum relationships. Here stratification is by the dependent variable "smoke" and within each smoking group education is modelled as a multinomial distribution conditional on sex, giving the correct model for `f7`.

```
R> ods <- ods[!is.na(ods$smoke), ] # remove 10 observations with missing "smoke"
R> s8 <- syn.strata(ods, m = 5, method = "parametric", strata = "smoke",
```

[1] From the expected value of a half-Normal distribution

```
+   seed = 5678, visit.sequence = c("smoke", "sex", "edu", "age"),
+   print.flag = FALSE, tab.strataobs = FALSE)
R> f8 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s8,
+   family = "binomial")
R> compare(f8, ods, plot.intercept = TRUE, plot = "coef")

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
    data = s8)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)         0.45921   0.4078  0.05142        0.47653     0.8784
sexFEMALE           1.03712   1.0592 -0.02211       -0.14727     0.9624
eduVOC              0.08504   0.1139 -0.02884       -0.22471     0.9427
eduSEC              0.44279   0.5074 -0.06457       -0.46064     0.8825
eduHIGH             0.90379   0.9864 -0.08262       -0.46585     0.8812
sexFEMALE:eduVOC   -0.59573  -0.6058  0.01004        0.05417     0.9862
sexFEMALE:eduSEC   -0.36813  -0.4497  0.08159        0.41637     0.8938
sexFEMALE:eduHIGH  -0.76002  -0.7986  0.03854        0.16529     0.9578

Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.9231
Mean absolute std. coef diff:  0.3014

Mahalanobis distance ratio for lack-of-fit (target 1.0): 0.49
Lack-of-fit test: 3.948; p-value 0.8618 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:
```

As expected, there is no evidence of any lack of fit and $|\bar{z}|$ is not out of line with what we expect with `m = 5`. The second example is more typical of what a user of **synthpop** might use with the default method `cart` for all variables.

```
R> ods <- ods[!is.na(ods$smoke), ] # remove 10 observationswith missing "smoke"
R> s9 <- syn(ods, m = 5, seed = 5678,
+   visit.sequence = c("sex", "edu", "age", "smoke"), print.flag = FALSE)
R> f9 <- glm.synds(smoke ~ sex  + age + edu + sex * age, data = s9,
+   family = "binomial")
R> compare(f9, ods)

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + age + edu + sex * age, family = "binomial",
    data = s9)
```

```
Differences between results based on synthetic and observed data:
              Synthetic  Observed      Diff Std. coef diff CI overlap
(Intercept)    0.197102  0.400261 -0.203160        -1.2463     0.6821
sexFEMALE      0.593279  0.224701  0.368578         2.0424     0.4790
age            0.008956  0.003104  0.005852         2.2534     0.4252
eduVOC        -0.030571 -0.050985  0.020414         0.2058     0.9475
eduSEC         0.448646  0.393783  0.054862         0.5315     0.8644
eduHIGH        0.659803  0.688391 -0.028588        -0.2374     0.9394
sexFEMALE:age -0.003031  0.007702 -0.010733        -2.9877     0.2378

Measures for 5 syntheses and 7 coefficients
Mean confidence interval overlap:  0.6536
Mean absolute std. coef diff:  1.358

Mahalanobis distance ratio for lack-of-fit (target 1.0): 10.15
Lack-of-fit test: 71.02; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 7 degrees of freedom.

Confidence interval plot:
```

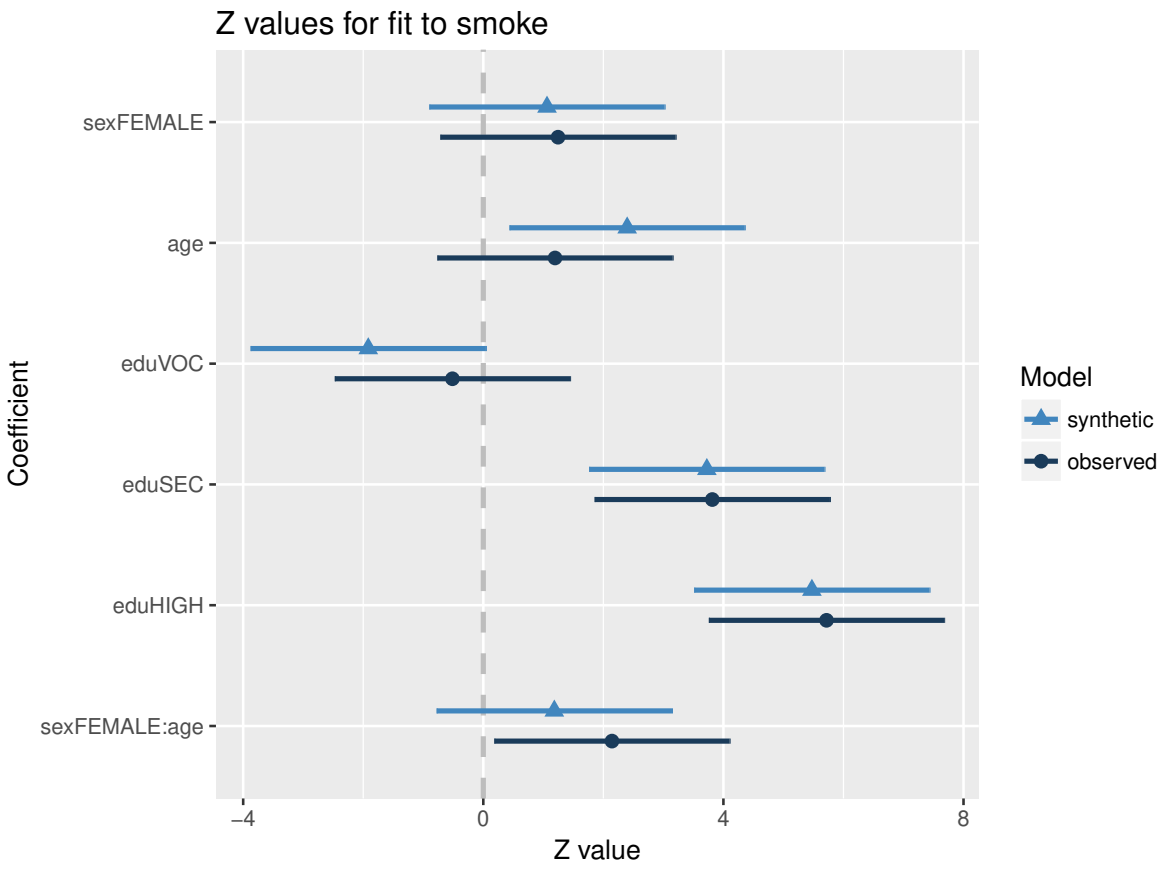

Figure 1: Comparison of intervals for fit `f8` and original data.

While the lack-of-fit test indicates differences and there is clear evidence of bias for two of the seven coefficients and some evidence for a third one, the general pattern of coefficients, illustrated in Figure 1, is the same fore both intervals and would not mislead any decisions to be made on the basis of preliminary analysis with the synthetic data. This is an example of a general point that we should not expect that these tests of fit should show no differences since it is sufficient for the synthetic data to be a good approximation to the original. The p-values tell us that there is evidence of a difference but not how important it may be. The effect size, measured here by z_j (`Std. coef diff`), gives a better measure of importance. In this example the largest absolute effect size is only `1.4`. The third synthesis uses the method `"sample"` for all variables, a bootstrap sample that does not preserve any of the relationships between variables, and the fitted model is clearly wrong with every coefficient giving evidence of a difference and the large absolute effect sizes, with three over `5`.

```
R> s10 <- syn(ods, m = 5, seed = 5678, method = "sample",
+   visit.sequence = c("sex", "edu", "age", "smoke"), print.flag = FALSE)
R> f10 <- glm.synds(smoke ~ sex  + age + edu + sex * edu, data = s10,
+   family = "binomial")
R> compare(f10, ods)

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s10)

Differences between results based on synthetic and observed data:
                   Synthetic  Observed      Diff Std. coef diff CI overlap
(Intercept)         1.102e+00  0.028147  1.073657          6.757   -0.72377
sexFEMALE          -1.165e-01  1.033275 -1.149825         -7.641   -0.94926
age                 9.724e-05  0.006355 -0.006258         -3.200    0.18369
eduVOC             -6.982e-02  0.219490 -0.289315         -2.182    0.44343
eduSEC              3.337e-02  0.613072 -0.579698         -4.020   -0.02552
eduHIGH            -2.542e-02  1.078920 -1.104339         -6.137   -0.56569
sexFEMALE:eduVOC    1.778e-01 -0.576577  0.754354          4.063   -0.03639
sexFEMALE:eduSEC    1.574e-02 -0.442560  0.458298          2.336    0.40406
sexFEMALE:eduHIGH   1.463e-01 -0.752495  0.898754          3.844    0.01931

Measures for 5 syntheses and 9 coefficients
Mean confidence interval overlap:  -0.1389
Mean absolute std. coef diff:  4.464

Mahalanobis distance ratio for lack-of-fit (target 1.0): 108.4
Lack-of-fit test: 975.3; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom.
```

```
Confidence interval plot:
```

When some variables in the model are not synthesised we need to set `incomplete = TRUE` for correct inference. This leads to the calculation of the variance matrix of the differences as $V_{diff} = B_m/m$. This variance matrix will be singular if the `m` is smaller than then number of coefficients (p) and `m` considerably larger than p is recommended. In this case the lack-of-fit test needs to use Hotelling's T^2 statistic referred to an F distribution with degrees of freedom p and $m - p$. In the next example two variables are unsynthesised so methods for incomplete/partial synthesis are used and m increased to 20.

```
R> s11 <- syn(ods, seed = 910011, m = 20, method = c("", "", "cart", "cart"),
+   print.flag = FALSE)
R> f11 <- glm.synds(smoke ~ sex + age + edu + sex * edu, data = s11,
+   family = "binomial")
R> compare(f11, ods)

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + age + edu + sex * edu, family = "binomial",
    data = s11)

Differences between results based on synthetic and observed data:
                  Synthetic  Observed       Diff Std. coef diff CI overlap
(Intercept)        0.039589  0.028147  1.144e-02        0.07201     0.9816
sexFEMALE          1.066542  1.033275  3.327e-02        0.22107     0.9436
age                0.006438  0.006355  8.294e-05        0.04241     0.9892
eduVOC             0.191791  0.219490 -2.770e-02       -0.20888     0.9467
eduSEC             0.571649  0.613072 -4.142e-02       -0.28726     0.9267
eduHIGH            1.148183  1.078920  6.926e-02        0.38493     0.9018
sexFEMALE:eduVOC  -0.559032 -0.576577  1.754e-02        0.09449     0.9759
sexFEMALE:eduSEC  -0.462752 -0.442560 -2.019e-02       -0.10292     0.9737
sexFEMALE:eduHIGH -0.951924 -0.752495 -1.994e-01       -0.85302     0.7824

Measures for 20 syntheses and 9 coefficients
Mean confidence interval overlap:  0.9357
Mean absolute std. coef diff:  0.2519

Mahalanobis distance ratio for lack-of-fit (target 1.0): 3.38
Lack-of-fit test: 30.42; p-value 4e-04 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom.

Confidence interval plot:
```

This gives a fit that, judged by the values of z_j, (`Std. coef diff`) and the overlaps is acceptable, but with `m = 20` there is evidence of lack-of-fit for two coefficients.

4.3 Confidence interval overlap when `population.inference = FALSE`

In this case confidence interval overlaps are calculated using the standard errors from the original fit. The overlap for the i^{th} coefficient is

$$IO_j = \left[\frac{min(u_o, u_s) - max(l_o, l_s)}{u_o - l_o}\right] \tag{1}$$

where the confidence interval for the original coefficient is (l_o, u_o) and for that from the synthesised data is (l_s, u_s). When the intervals do not overlap IO_j becomes negative. Note that this interval is a simplification of Equation 2, below, proposed by [3] because both intervals are the same length. The two intervals are offset by $|z_j|$ so that standardised difference and the interval overlap are linearly related by $IO_j = 1 - |z_j|/(2q_{N(1-\alpha/2)})$, where $q_{N(1-\alpha/2)}$ is the quantile of the Normal distribution used in calculating the confidence intervals, i.e. 1.96 for 95% intervals. The average of the overlaps can then be used as a second summary measure of utility.

The expected value of IO_j for 95% intervals will be $1 - \sqrt{2/(m\pi)}/(2 \cdot 1.96)$, giving expected overlaps of 79.6%, 90.9%, 93.5% and 98.6% for $m = 1, 5, 10, 200$.

To illustrate these methods we select three different fitted models. In the first case the synthesis model is compatible with the model being fitted. A model with smoking predicted from sex, education and its interactions is evaluated. The synthesis is stratified by the dependent variable, smoking, with missing values removed and a parametric model with sex and education is fitted in each stratum, giving a synthesis compatible with the model.

```
R> ods <- ods[!is.na(ods$smoke), ]
R> s12 <- syn.strata(ods, m = 5, visit.sequence = c(4, 1, 2),
+   method = "parametric", strata = "smoke", seed = 5678, print.flag = FALSE)

Number of observations in strata (original data):
 YES   NO
1277 3713

R> s13 <- syn.strata(ods, m = 5, visit.sequence = c(4, 1, 2),
+   method = "parametric", strata = "smoke", seed = 1234, proper = TRUE,
+   print.flag = FALSE, tab.strataobs = FALSE)
R> f12 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s12,
+   family = "binomial")
R> compare(f12, ods, plot.intercept = TRUE, plot = "coef")

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
```

```
    data = s12)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)         0.45676   0.4078  0.04897         0.4538     0.8842
sexFEMALE           1.01348   1.0592 -0.04574        -0.3047     0.9223
eduVOC              0.09649   0.1139 -0.01739        -0.1355     0.9654
eduSEC              0.48782   0.5074 -0.01953        -0.1393     0.9645
eduHIGH             0.94450   0.9864 -0.04191        -0.2363     0.9397
sexFEMALE:eduVOC   -0.55190  -0.6058  0.05387         0.2908     0.9258
sexFEMALE:eduSEC   -0.48621  -0.4497 -0.03649        -0.1862     0.9525
sexFEMALE:eduHIGH  -0.74999  -0.7986  0.04858         0.2083     0.9469

Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.9377
Mean absolute std. coef diff:  0.2444

Mahalanobis distance ratio for lack-of-fit (target 1.0): 0.7
Lack-of-fit test: 5.57; p-value 0.6953 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:

R> f13 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s13,
+   family = "binomial")
R> compare(f13, ods, plot.intercept = TRUE)

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
    data = s13)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)          0.2855   0.4078 -0.12230       -1.13337     0.7109
sexFEMALE            1.2375   1.0592  0.17828        1.18760     0.6970
eduVOC               0.2531   0.1139  0.13917        1.08434     0.7234
eduSEC               0.6540   0.5074  0.14666        1.04631     0.7331
eduHIGH              0.9508   0.9864 -0.03564       -0.20096     0.9487
sexFEMALE:eduVOC    -0.8695  -0.6058 -0.26370       -1.42346     0.6369
sexFEMALE:eduSEC    -0.5814  -0.4497 -0.13170       -0.67214     0.8285
sexFEMALE:eduHIGH   -0.7774  -0.7986  0.02114        0.09067     0.9769

Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.7819
Mean absolute std. coef diff:  0.8549

Mahalanobis distance ratio for lack-of-fit (target 1.0): 2.76
```

```
Lack-of-fit test: 22.12; p-value 0.0047 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:
```

The first synthesis gives results in line with what would be expected for standardised differences and overlaps from a correct model with `m = 5`, and the second with synthesis from the posterior predictive distribution of the parameters gives larger standardised differences and lower overlaps. In neither case does the lack-of-fit test suggest any problem with the synthesising model.

The next example uses parametric models which capture some of the relationships but largely miss the interaction between education and sex.

```
R> s14 <- syn(ods, m = 5, seed = 9101112, method = "parametric",
+   print.flag = FALSE)
R> s15 <- syn(ods, m = 5, seed = 1415, method = "parametric",
+   proper = TRUE, print.flag = FALSE)
R> f14 <- glm.synds(smoke ~ sex  + edu + age + sex * edu, data = s14,
+   family = "binomial")
R> compare(f13, ods, plot.intercept = TRUE, plot = "coef")

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
    data = s13)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)          0.2855   0.4078 -0.12230       -1.13337     0.7109
sexFEMALE            1.2375   1.0592  0.17828        1.18760     0.6970
eduVOC               0.2531   0.1139  0.13917        1.08434     0.7234
eduSEC               0.6540   0.5074  0.14666        1.04631     0.7331
eduHIGH              0.9508   0.9864 -0.03564       -0.20096     0.9487
sexFEMALE:eduVOC    -0.8695  -0.6058 -0.26370       -1.42346     0.6369
sexFEMALE:eduSEC    -0.5814  -0.4497 -0.13170       -0.67214     0.8285
sexFEMALE:eduHIGH   -0.7774  -0.7986  0.02114        0.09067     0.9769

Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.7819
Mean absolute std. coef diff:  0.8549

Mahalanobis distance ratio for lack-of-fit (target 1.0): 2.76
Lack-of-fit test: 22.12; p-value 0.0047 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:
```

```
R> f15 <- glm.synds(smoke ~ sex  + edu + age + sex * edu, data = s15,
+   family = "binomial")
R> compare(f15, ods, plot.intercept = TRUE)

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + age + sex * edu, family = "binomial",
    data = s15)

Differences between results based on synthetic and observed data:
                  Synthetic  Observed      Diff Std. coef diff CI overlap
(Intercept)       -0.094687  0.028147 -0.122834        -0.7731     0.8028
sexFEMALE          0.659187  1.033275 -0.374088        -2.4859     0.3658
eduVOC             0.233921  0.219490  0.014431         0.1088     0.9722
eduSEC             0.674266  0.613072  0.061194         0.4244     0.8917
eduHIGH            0.961463  1.078920 -0.117457        -0.6528     0.8335
age                0.008807  0.006355  0.002451         1.2534     0.6802
sexFEMALE:eduVOC  -0.031396 -0.576577  0.545180         2.9361     0.2510
sexFEMALE:eduSEC  -0.003785 -0.442560  0.438775         2.2365     0.4294
sexFEMALE:eduHIGH -0.186075 -0.752495  0.566420         2.4227     0.3819

Measures for 5 syntheses and 9 coefficients
Mean confidence interval overlap:  0.6232
Mean absolute std. coef diff:  1.477

Mahalanobis distance ratio for lack-of-fit (target 1.0): 10.71
Lack-of-fit test: 96.42; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom.

Confidence interval plot:
```

Both syntheses identify the problems with the synthesis model, but the first analysis with `proper = FALSE` shows it more clearly.

4.4 Confidence intervals for `population.inference = TRUE`.

When the parameter `population.inference` of the `compare.fit.synds()` function is set to `TRUE` interval overlaps are calculated as

$$IO_j = 0.5\left[\frac{min(u_o, u_s) - max(l_o, l_s)}{u_o - l_o} + \frac{min(u_o, u_s) - max(l_o, l_s)}{u_s - l_s}\right] \tag{2}$$

as proposed by [3]. Here the proportion of overlap is calculated with the geometric mean of the interval lengths in the denominator. When the intervals are calculated on the original scales

(plot = "coef") they differ in length, with that from the synthetic data usually being wider. If the intervals are printed and plotted as z statistics (plot = "z") these differences in width of intervals are not apparent because each one has been standardised by its standard error. The differences are most pronounced for small m and when the data have been synthesised with proper = TRUE as in s16 below.

```
R> s16 <- syn(ods, proper = TRUE, print.flag = FALSE)
R> f16 <- glm.synds(smoke ~ sex  + edu  + sex * edu, data = s16,
+   family = "binomial")
R> compare(f16, ods, plot.intercept = TRUE, plot = "coef")

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
    data = s16)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)         0.36822   0.4078 -0.03957        -0.3667     0.9064
sexFEMALE           1.24320   1.0592  0.18397         1.2255     0.6874
eduVOC              0.01954   0.1139 -0.09434        -0.7350     0.8125
eduSEC              0.74137   0.5074  0.23401         1.6695     0.5741
eduHIGH             1.10822   0.9864  0.12182         0.6869     0.8248
sexFEMALE:eduVOC   -0.56713  -0.6058  0.03863         0.2085     0.9468
sexFEMALE:eduSEC   -0.81774  -0.4497 -0.36803        -1.8782     0.5208
sexFEMALE:eduHIGH  -1.18534  -0.7986 -0.38677        -1.6586     0.5769

Measures for one synthesis and 8 coefficients
Mean confidence interval overlap:  0.7312
Mean absolute std. coef diff:  1.054

Mahalanobis distance ratio for lack-of-fit (target 1.0): 1.65
Lack-of-fit test: 13.18; p-value 0.1058 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:

R> compare(f16, ods, plot.intercept = TRUE, population.inference = TRUE,
+   plot = "coef")

Call used to fit models to the data:
glm.synds(formula = smoke ~ sex + edu + sex * edu, family = "binomial",
    data = s16)

Differences between results based on synthetic and observed data:
                  Synthetic Observed     Diff Std. coef diff CI overlap
(Intercept)         0.36822   0.4078 -0.03957        -0.3667     0.7887
```

```
sexFEMALE           1.24320   1.0592  0.18397         1.2255     0.7887
eduVOC              0.01954   0.1139 -0.09434        -0.7350     0.7887
eduSEC              0.74137   0.5074  0.23401         1.6695     0.7414
eduHIGH             1.10822   0.9864  0.12182         0.6869     0.7887
sexFEMALE:eduVOC   -0.56713  -0.6058  0.03863         0.2085     0.7887
sexFEMALE:eduSEC   -0.81774  -0.4497 -0.36803        -1.8782     0.6995
sexFEMALE:eduHIGH  -1.18534  -0.7986 -0.38677        -1.6586     0.7436

Measures for one synthesis and 8 coefficients
Mean confidence interval overlap:  0.766
Mean absolute std. coef diff:  1.054

Mahalanobis distance ratio for lack-of-fit (target 1.0): 1.65
Lack-of-fit test: 13.18; p-value 0.1058 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom.

Confidence interval plot:
```

5 Acknowledgement

We are most grateful to Dr. Jörg Drechsler for his constructive criticisms of a previous draft of this vignette and to other `synthpop` users for their feedback.

References

[1] Drechsler, J. *Synthetic Data Sets for Statistical Disclosure Control*. Springer, New York, 2011.

[2] Drechsler, J., and Reiter, J. P. An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. *Computational Statistics and Data Analysis 55* (2011), 3232–3243.

[3] Karr, A., Kohnen, C. N., Organian, A., Reiter, J. P., and Sanil, A. P. A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician 60*, 3 (2006), 224–232.

[4] Nowok, B., Raab, G. M., and Dibben, C. synthpop : Bespoke creation of synthetic data in R. *Journal of Statistical Software 74* (2016), 1–26. Available at `https://www.jstatsoft.org/article/view/v074i11`.

[5] Raab, G. M.; Nowok, B., and Dibben, C. Practical data synthesis for large samples. *Journal of Privacy and Confidentiality 7*, 3 (2016–2017), 67–97. Available at `http://repository.cmu.edu/jpc/vol7/iss3/4`.

[6] Raghunathan, T. E., Reiter, J. P., and Rubin, D. B. Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics 19*, 1 (2003), 1–17.

[7] Reiter, J. P. Inference for partially synthetic, public use microdata sets. *Survey Methodology 29* (2003), 181–188.

[8] Woo, M.-J., Reiter, J. P., Oganian, A., and Karr, A. F. Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality 1* (2009), 111–124.