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 ($\texttt{m = 1}$) of the same size as the original ($\texttt{k = n}$) that can be analysed as if it were the original data. In this situation an analysis of the synthetic data set, as if it were the original, is all that is required and 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 ≠ n`) the standard errors of the coefficients from the synthetic fit require adjustment to estimate the results that would be obtained from the real 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 we need to use estimators that 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 population parameters.

The contribution from the first source decreases as the number of synthetic data sets increases, while that from the second source is unchanged. 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()` and `lm.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. The user can also specify that methods proposed for partially synthetic data [7] should be used to calculate standard errors by (parameter `incomplete` of the function `summary.fit.synds()` or `compare.fit.synds()`). Although these are valid methods for any type of synthetic data they have the disadvantage of requiring multiple syntheses and should only be used when required, as discussed below. 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 ith 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
 -2.010   0.527   0.638   0.794   1.043

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.25448    0.16345    1.56  0.11949
sexFEMALE           0.97619    0.15858    6.16  7.5e-10 ***
age                 0.00442    0.00196    2.26  0.02386 *
eduVOC              0.13825    0.13866    1.00  0.31873
eduSEC              0.66224    0.15053    4.40  1.1e-05 ***
eduHIGH             0.95747    0.18105    5.29  1.2e-07 ***
sexFEMALE:eduVOC   -0.67357    0.19178   -3.51  0.00044 ***
sexFEMALE:eduSEC   -0.64464    0.20224   -3.19  0.00144 **
sexFEMALE:eduHIGH  -0.63669    0.23833   -2.67  0.00755 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5576.6  on 4979  degrees of freedom
Residual deviance: 5413.0  on 4971  degrees of freedom
  (20 observations deleted due to missingness)
AIC: 5431

Number of Fisher Scoring iterations: 4

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

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with a single synthesis.
Inference to coefficients and standard errors that
would be obtained from the observed 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)         0.25448       0.16345    1.56        0.11949
sexFEMALE           0.97619       0.15858    6.16        7.5e-10 ***
age                 0.00442       0.00196    2.26        0.02386 *
eduVOC              0.13825       0.13866    1.00        0.31873
eduSEC              0.66224       0.15053    4.40        1.1e-05 ***
eduHIGH             0.95747       0.18105    5.29        1.2e-07 ***
sexFEMALE:eduVOC   -0.67357       0.19178   -3.51        0.00044 ***
sexFEMALE:eduSEC   -0.64464       0.20224   -3.19        0.00144 **
sexFEMALE:eduHIGH  -0.63669       0.23833   -2.67        0.00755 **
---
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 methods proposed for partially synthetic data [7] are used by setting parameter `incomplete = TRUE` in `summary.fit.synds()`. One small advantage of using `glm.synds()` is that it checks and gives warnings if the model includes unsynthesised variables in such a way that they would invalidate the inference, 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"))

STERN WARNING: Variable(s) in formula are
not in visit sequence: sex
*****************************************
This inference will be wrong because these
variables will not match synthesised data
*****************************************

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with a single synthesis.
Inference to coefficients and standard errors that
would be obtained from the observed 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)         0.32188       0.16208    1.99         0.0470 *
sexFEMALE           0.06425       0.14333    0.45         0.6540
age                 0.00992       0.00196    5.06       4.2e-07 ***
eduVOC              0.15796       0.14240    1.11         0.2673
eduSEC              0.46799       0.14835    3.15         0.0016 **
eduHIGH             0.88506       0.17394    5.09       3.6e-07 ***
sexFEMALE:eduVOC   -0.16701       0.18116   -0.92         0.3566
sexFEMALE:eduSEC   -0.06650       0.19229   -0.35         0.7295
sexFEMALE:eduHIGH  -0.08091       0.22691   -0.36         0.7214
---
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> # original data
R> summary(glm(smoke ~ sex + age + edu + sex * edu,
+    data = ods, family = "binomial"))

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

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.026  -1.295   0.634   0.808   1.123

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.02815    0.15889    0.18   0.8594
sexFEMALE            1.03328    0.15048    6.87 6.6e-12 ***
age                  0.00636    0.00196    3.25  0.0012 **
eduVOC               0.21949    0.13261    1.66  0.0979 .
eduSEC               0.61307    0.14420    4.25 2.1e-05 ***
eduHIGH              1.07892    0.17994    6.00 2.0e-09 ***
sexFEMALE:eduVOC    -0.57658    0.18568   -3.11  0.0019 **
sexFEMALE:eduSEC    -0.44256    0.19619   -2.26  0.0241 *
sexFEMALE:eduHIGH   -0.75249    0.23379   -3.22  0.0013 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5669.7  on 4982  degrees of freedom
Residual deviance: 5478.1  on 4974  degrees of freedom
  (17 observations deleted due to missingness)
AIC: 5496

Number of Fisher Scoring iterations: 4
```

```
R> # synthetic data
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
-1.970  -1.309   0.694   0.772   1.058

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.22926    0.50643    0.45     0.65
sexFEMALE            0.55602    0.45137    1.23     0.22
age                  0.00349    0.00625    0.56     0.58
eduVOC               0.24190    0.42574    0.57     0.57
eduSEC               0.75265    0.47261    1.59     0.11
eduHIGH              0.74131    0.52771    1.40     0.16
sexFEMALE:eduVOC     0.07160    0.57665    0.12     0.90
sexFEMALE:eduSEC    -0.00858    0.61475   -0.01     0.99
sexFEMALE:eduHIGH   -0.20692    0.70009   -0.30     0.77

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 563.36  on 497  degrees of freedom
Residual deviance: 549.39  on 489  degrees of freedom
  (2 observations deleted due to missingness)
AIC: 567.4

Number of Fisher Scoring iterations: 4

R> # estimating original from synthetic
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+    data = s3, family = "binomial"))

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with a single synthesis.
Inference to coefficients and standard errors that
would be obtained from the observed 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)         0.22926       0.16015    1.43          0.152
sexFEMALE           0.55602       0.14274    3.90        9.8e-05 ***
age                 0.00349       0.00198    1.77          0.077 .
eduVOC              0.24190       0.13463    1.80          0.072 .
eduSEC              0.75265       0.14945    5.04        4.8e-07 ***
eduHIGH             0.74131       0.16688    4.44        8.9e-06 ***
sexFEMALE:eduVOC    0.07160       0.18235    0.39          0.695
sexFEMALE:eduSEC   -0.00858       0.19440   -0.04          0.965
sexFEMALE:eduHIGH  -0.20692       0.22139   -0.93          0.350
---
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 no coefficients appearing significant, whereas `glm.synds()` estimates what would be found from the original. 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

The default setting of the parameter `incomplete` in `summary.fit.synds()` is `FALSE` indicating that methods of inference derived for the case when all of the original data have been replaced by synthesised values are used. This code gets inference to population parameters for our example in this case.

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

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with a single synthesis.
Inference to population coefficients.

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)          0.25448      0.23115  1.10      0.27093
sexFEMALE            0.97619      0.22426  4.35      1.3e-05 ***
age                  0.00442      0.00277  1.60      0.11012
eduVOC               0.13825      0.19610  0.71      0.48079
eduSEC               0.66224      0.21288  3.11      0.00187 **
eduHIGH              0.95747      0.25604  3.74      0.00018 ***
sexFEMALE:eduVOC    -0.67357      0.27122 -2.48      0.01301 *
sexFEMALE:eduSEC    -0.64464      0.28602 -2.25      0.02421 *
sexFEMALE:eduHIGH   -0.63669      0.33705 -1.89      0.05889 .
---
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 $\sqrt{v_m/m + \bar{v}_m k/n)}$. 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)

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with a single synthesis.
Inference to population coefficients.

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)             0.15461      0.27485  0.56      0.57376
sexFEMALE               1.17984      0.26838  4.40      1.1e-05 ***
age                     0.00417      0.00341  1.23      0.22049
eduVOC                  0.28499      0.22918  1.24      0.21368
eduSEC                  0.64016      0.24774  2.58      0.00977 **
eduHIGH                 1.19732      0.32832  3.65      0.00027 ***
sexFEMALE:eduVOC       -0.71180      0.32866 -2.17      0.03033 *
sexFEMALE:eduSEC       -0.51439      0.34872 -1.48      0.14019
sexFEMALE:eduHIGH      -1.11301      0.42441 -2.62      0.00873 **
---
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)

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with 10 syntheses.
Inference to population coefficients.

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)        0.15432     0.17512  0.88       0.3782
sexFEMALE          0.95188     0.16603  5.73     9.9e-09 ***
age                0.00600     0.00216  2.78       0.0055 **
eduVOC             0.17074     0.14605  1.17       0.2424
eduSEC             0.49031     0.15796  3.10       0.0019 **
eduHIGH            0.87320     0.19536  4.47     7.8e-06 ***
sexFEMALE:eduVOC  -0.53322     0.20491 -2.60       0.0093 **
sexFEMALE:eduSEC  -0.32589     0.21636 -1.51       0.1320
sexFEMALE:eduHIGH -0.50287     0.25758 -1.95       0.0509 .
---
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.

If only the unchanged variables are all predictor variables in a statistical model then the methods for population inference described above will still apply. However, if one of the unchanged variables is the dependent variable in a statistical model the results above will not hold because the between-synthesis variance will generally be smaller than it would have been if the variable had been completely synthesised. In this case 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. This estimate of standard error is valid for any inference from synthetic data, but the requirement for multiple syntheses, and the associated increase in disclosure risk, means it should only be used when it is the only choice. Simulations and theoretical results have confirmed that it is only required for population inference when the dependent variable is not completely synthesised.

To use the method due to [7] the parameter `incomplete` must be set to `TRUE`. In the next example `smoke` is not synthesised and the other variables are synthesised from its unchanged values.

```
R> s6 <- syn(ods, seed = 910011, m = 6,
+   method = c("", "cart", "cart", "cart"), print.flag = FALSE)
R> summary(glm.synds(smoke ~ sex + age + edu + sex * edu,
+   data = s6, family = "binomial"), population.inference = TRUE,
+   incomplete = TRUE)

Note: Your response variable is not synthesised. The compare
method for evaluating lack-of-fit and a summary of your model
with population.inference = TRUE should use incomplete = TRUE
(see vignette on inference for details).

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with 6 syntheses.
Inference to population coefficients.

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)        -0.04101     0.16765 -0.24     0.80677
sexFEMALE           1.09972     0.15641  7.03     2.0e-12 ***
age                 0.00669     0.00206  3.25     0.00117 **
eduVOC              0.26191     0.13921  1.88     0.05992 .
eduSEC              0.67958     0.15291  4.44     8.8e-06 ***
eduHIGH             1.18870     0.19202  6.19     6.0e-10 ***
sexFEMALE:eduVOC   -0.61950     0.18993 -3.26     0.00111 **
sexFEMALE:eduSEC   -0.50483     0.21715 -2.32     0.02008 *
sexFEMALE:eduHIGH  -0.90513     0.23823 -3.80     0.00015 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

Note: Your response variable is not synthesised. The compare
method for evaluating lack-of-fit and a summary of your model
with population.inference = TRUE should use incomplete = TRUE
(see vignette on inference for details).

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with 6 syntheses.
Inference to population coefficients.

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)        -0.04101     0.17140 -0.24     0.81092
sexFEMALE           1.09972     0.16233  6.77     1.2e-11 ***
age                 0.00669     0.00212  3.15     0.00161 **
eduVOC              0.26191     0.14316  1.83     0.06733 .
eduSEC              0.67958     0.15537  4.37     1.2e-05 ***
eduHIGH             1.18870     0.19391  6.13     8.8e-10 ***
sexFEMALE:eduVOC   -0.61950     0.20065 -3.09     0.00202 **
sexFEMALE:eduSEC   -0.50483     0.21196 -2.38     0.01723 *
sexFEMALE:eduHIGH  -0.90513     0.25270 -3.58     0.00034 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The first command here gives the correct estimates for incompletely synthesised data with standard errors calculated from differences between the syntheses. The second one gives the

results you would obtain by ignoring the fact that some of the variables have not been synthesised. Differences are very small here because with `m = 6` the contribution from the first term under the square root is small compared to the second term.

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)

Note: Your response variable is not synthesised. The compare
method for evaluating lack-of-fit and a summary of your model
with population.inference = TRUE should use incomplete = TRUE
(see vignette on inference for details).

Warning: Note that all these results depend on the synthesis model being correct.

Fit to synthetic data set with 6 syntheses.
Inference to population coefficients.

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)         -0.04101     0.16765 -0.24      0.80677
sexFEMALE            1.09972     0.15641  7.03      2.0e-12 ***
age                  0.00669     0.00206  3.25      0.00117 **
eduVOC               0.26191     0.13921  1.88      0.05992 .
eduSEC               0.67958     0.15291  4.44      8.8e-06 ***
eduHIGH              1.18870     0.19202  6.19      6.0e-10 ***
sexFEMALE:eduVOC    -0.61950     0.18993 -3.26      0.00111 **
sexFEMALE:eduSEC    -0.50483     0.21715 -2.32      0.02008 *
sexFEMALE:eduHIGH   -0.90513     0.23823 -3.80      0.00015 ***
---
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)       0.105834 -0.204342
sexFEMALE         0.925590  1.096909
```

```
age                    0.006438   0.007549
eduVOC                 0.186651   0.390548
eduSEC                 0.561233   0.738674
eduHIGH                1.032438   1.474124
sexFEMALE:eduVOC      -0.566410  -0.574549
sexFEMALE:eduSEC      -0.387106  -0.315839
sexFEMALE:eduHIGH     -0.716155  -1.011531


z values:
                       syn=1   syn=2
(Intercept)           0.6775  -1.295
sexFEMALE             6.2021   7.581
age                   3.3170   3.825
eduVOC                1.4097   3.001
eduSEC                3.8781   5.283
eduHIGH               5.8812   7.741
sexFEMALE:eduVOC     -3.0815  -3.148
sexFEMALE:eduSEC     -1.9765  -1.645
sexFEMALE:eduHIGH    -3.0885  -4.200
```

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 the 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

---

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

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> s7 <- syn.strata(ods, m = 5, method = "parametric", strata = "smoke",
+   seed = 5678, visit.sequence = c("smoke", "sex", "edu", "age"),
+   print.flag = FALSE, tab.strataobs = FALSE)
R> f7 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s7,
+   family = "binomial")
R> compare(f7, 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 = s7)


Differences between results based on synthetic and observed data:
                 Std. coef diff p value CI overlap
(Intercept)             0.32869   0.462     0.9161
sexFEMALE               0.02840   0.949     0.9928
eduVOC                 -0.05738   0.898     0.9854
eduSEC                 -0.55124   0.218     0.8594
eduHIGH                 0.28253   0.528     0.9279
sexFEMALE:eduVOC       -0.30147   0.500     0.9231
sexFEMALE:eduSEC        0.19246   0.667     0.9509
sexFEMALE:eduHIGH      -0.05116   0.909     0.9869


Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.9428
```

```
Mean absolute std. coef diff:  0.2242
Lack-of-fit: 7.942; p-value 0.439 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> s8 <- syn(ods, m = 5, seed = 5678,
+   visit.sequence = c("sex", "edu", "age", "smoke"), print.flag = FALSE)
R> f8 <- glm.synds(smoke ~ sex  + age + edu + sex * age, data = s8,
+   family = "binomial")
R> compare(f8, ods)

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

Differences between results based on synthetic and observed data:
            Std. coef diff p value CI overlap
(Intercept)        -0.42149   0.346     0.8925
sexFEMALE           1.21259   0.007     0.6907
age                 1.33064   0.003     0.6605
eduVOC             -0.45924   0.304     0.8828
eduSEC             -0.01688   0.970     0.9957
eduHIGH            -0.14326   0.749     0.9635
sexFEMALE:age      -1.88734   0.000     0.5185


Measures for 5 syntheses and 7 coefficients
Mean confidence interval overlap:  0.8006
Mean absolute std. coef diff:  0.7816
Lack-of-fit: 31.85; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 7 degrees of freedom
```

Confidence interval plot:

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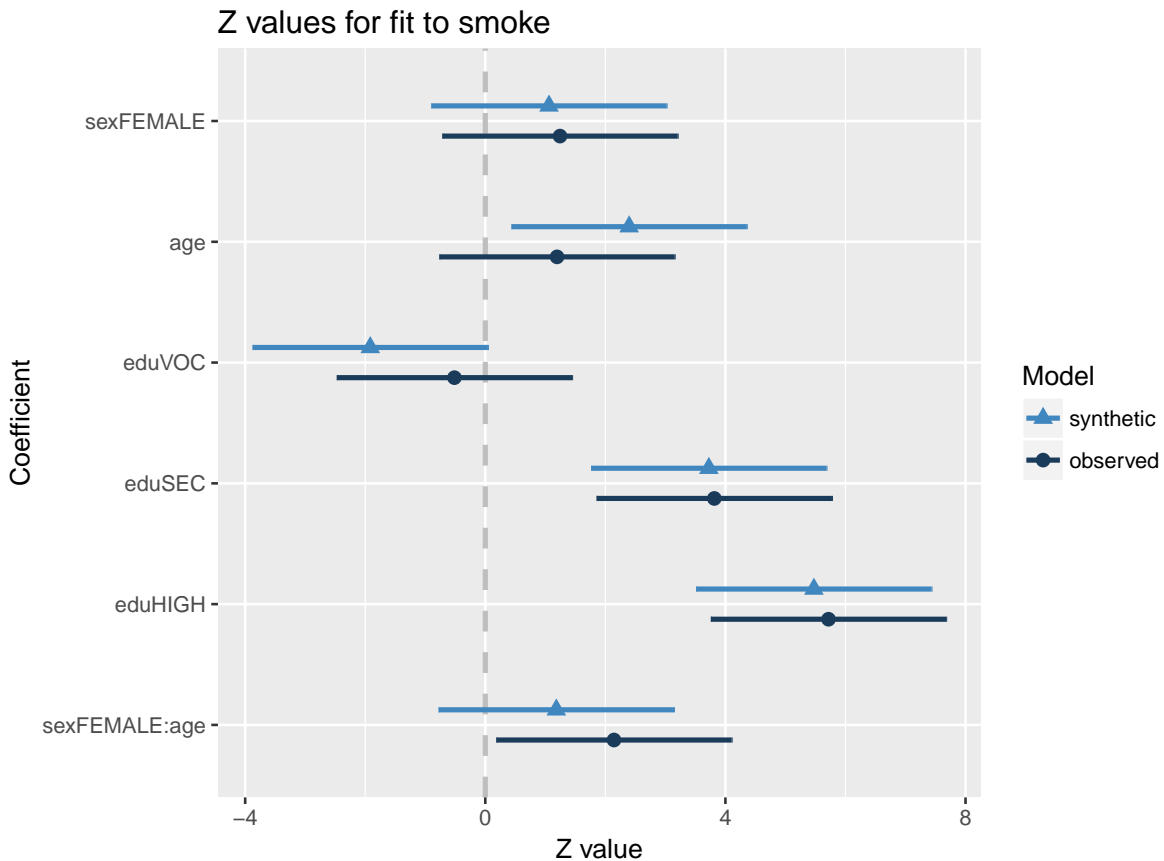
Figure 1: Comparison of intervals for fit `f8` and original data.

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> s9 <- syn(ods, m = 5, seed = 5678, method = "sample",
+   visit.sequence = c("sex", "edu", "age", "smoke"), print.flag = FALSE)
R> f9 <- glm.synds(smoke ~ sex  + age + edu + sex * edu, data = s9,
+   family = "binomial")
R> compare(f9, ods)

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

```
Differences between results based on synthetic and observed data:
                 Std. coef diff p value CI overlap
(Intercept)              6.4247   0.000   -0.63899
sexFEMALE               -5.9487   0.000   -0.51755
age                     -3.5460   0.000    0.09538
eduVOC                  -0.8268   0.065    0.78909
eduSEC                  -3.8452   0.000    0.01906
eduHIGH                 -5.8503   0.000   -0.49245
sexFEMALE:eduVOC         1.9332   0.000    0.50682
sexFEMALE:eduSEC         2.0516   0.000    0.47661
sexFEMALE:eduHIGH        2.5434   0.000    0.35116


Measures for 5 syntheses and 9 coefficients
Mean confidence interval overlap:  0.06546
Mean absolute std. coef diff:  3.663
Lack-of-fit: 941.1; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom

Confidence interval plot:
```

When the dependent variable is 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, including the dependent variable are unsynthesised so `incomplete = TRUE` is required and m increased to 20.

```
R> s10 <- syn(ods, seed = 910011, m = 20, method = c("", "", "cart", "cart"),
+    print.flag = FALSE)
R> f10 <- glm.synds(smoke ~ sex + age + edu + sex * edu, data = s10,
+    family = "binomial")

Note: Your response variable is not synthesised. The compare
method for evaluating lack-of-fit and a summary of your model
with population.inference = TRUE should use incomplete = TRUE
(see vignette on inference for details).

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:
                 Std. coef diff p value CI overlap
```

```
(Intercept)                   0.4368   0.051      0.8886
sexFEMALE                     0.1818   0.416      0.9536
age                          -0.3999   0.074      0.8980
eduVOC                       -0.3535   0.114      0.9098
eduSEC                       -0.3424   0.126      0.9127
eduHIGH                      -0.1115   0.618      0.9716
sexFEMALE:eduVOC              0.1105   0.621      0.9718
sexFEMALE:eduSEC             -0.3167   0.157      0.9192
sexFEMALE:eduHIGH            -0.2363   0.291      0.9397


Measures for 20 syntheses and 9 coefficients
Mean confidence interval overlap:  0.9294
Mean absolute std. coef diff:  0.2766
Lack-of-fit: 19.97; p-value 0.018 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom


Confidence interval plot:


R> compare(f10, ods, incomplete = TRUE)


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:
                Std. coef diff p value CI overlap
(Intercept)                   0.4368   0.017      0.8886
sexFEMALE                     0.1818   0.425      0.9536
age                          -0.3999   0.043      0.8980
eduVOC                       -0.3535   0.043      0.9098
eduSEC                       -0.3424   0.146      0.9127
eduHIGH                      -0.1115   0.567      0.9716
sexFEMALE:eduVOC              0.1105   0.622      0.9718
sexFEMALE:eduSEC             -0.3167   0.228      0.9192
sexFEMALE:eduHIGH            -0.2363   0.351      0.9397


Measures for 20 syntheses and 9 coefficients
Mean confidence interval overlap:  0.9294
Mean absolute std. coef diff:  0.2766
Lack-of-fit: 3.302; p-value 0.033 for test that synthesis model is compatible
with an F distribution with 9 and 11 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 ith 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> s11 <- 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> s12 <- 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> f11 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s11,
+   family = "binomial")
R> compare(f11, 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 = s11)


Differences between results based on synthetic and observed data:
                Std. coef diff p value CI overlap
(Intercept)              -0.1171    0.794     0.9701
sexFEMALE                 0.1947    0.663     0.9503
eduVOC                    0.3765    0.400     0.9040
eduSEC                   -0.2325    0.603     0.9407
eduHIGH                   0.4476    0.317     0.8858
sexFEMALE:eduVOC         -0.4413    0.324     0.8874
sexFEMALE:eduSEC          0.2850    0.524     0.9273
sexFEMALE:eduHIGH        -0.1582    0.724     0.9596


Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.9282
Mean absolute std. coef diff:  0.2816
Lack-of-fit: 6.581; p-value 0.582 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom


Confidence interval plot:


R> f12 <- glm.synds(smoke ~ sex  + edu + sex * edu, data = s12,
+   family = "binomial")
R> compare(f12, ods, plot.intercept = TRUE)


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:
                Std. coef diff p value CI overlap
(Intercept)               0.3933    0.534     0.8997
sexFEMALE                 0.2501    0.693     0.9362
eduVOC                   -0.4516    0.475     0.8848
eduSEC                    0.1152    0.855     0.9706
eduHIGH                  -0.5143    0.416     0.8688
sexFEMALE:eduVOC         -0.5270    0.405     0.8655
sexFEMALE:eduSEC         -0.6692    0.290     0.8293
sexFEMALE:eduHIGH         0.2907    0.646     0.9258


Measures for 5 syntheses and 8 coefficients
Mean confidence interval overlap:  0.8976
Mean absolute std. coef diff:  0.4014
Lack-of-fit: 6.462; p-value 0.596 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> s13 <- syn(ods, m = 5, seed = 9101112, method = "parametric",
+   print.flag = FALSE)
R> s14 <- syn(ods, m = 5, seed = 1415, method = "parametric",
+   proper = TRUE, print.flag = FALSE)
R> f13 <- glm.synds(smoke ~ sex  + edu + age + sex * edu, data = s13,
+   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 + age + sex * edu, family = "binomial",
    data = s13)


Differences between results based on synthetic and observed data:
                  Std. coef diff p value CI overlap
(Intercept)              -0.1239   0.782     0.9684
sexFEMALE                -2.2413   0.000     0.4282
eduVOC                   -0.1374   0.759     0.9649
eduSEC                    0.1301   0.771     0.9668
eduHIGH                  -0.8849   0.048     0.7742
age                       0.5695   0.203     0.8547
sexFEMALE:eduVOC          2.7262   0.000     0.3045
sexFEMALE:eduSEC          1.6872   0.000     0.5696
sexFEMALE:eduHIGH         2.7229   0.000     0.3054


Measures for 5 syntheses and 9 coefficients
Mean confidence interval overlap:  0.6819
Mean absolute std. coef diff:  1.247
Lack-of-fit: 81.3; p-value 0 for test that synthesis model is compatible
with a chi-squared test with 9 degrees of freedom


Confidence interval plot:

R> f14 <- glm.synds(smoke ~ sex  + edu + age + sex * edu, data = s14,
+   family = "binomial")
R> compare(f14, ods, plot.intercept = TRUE)

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

```
   data = s14)

Differences between results based on synthetic and observed data:
               Std. coef diff p value CI overlap
(Intercept)            -0.46958   0.458     0.8802
sexFEMALE              -2.42360   0.000     0.3817
eduVOC                 -0.57514   0.363     0.8533
eduSEC                 -0.02518   0.968     0.9936
eduHIGH                -1.45091   0.022     0.6299
age                     1.14444   0.070     0.7080
sexFEMALE:eduVOC        3.27967   0.000     0.1633
sexFEMALE:eduSEC        2.12833   0.001     0.4570
sexFEMALE:eduHIGH       3.25245   0.000     0.1703


Measures for 5 syntheses and 9 coefficients
Mean confidence interval overlap:  0.5819
Mean absolute std. coef diff:  1.639
Lack-of-fit: 52.33; 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> s15 <- syn(ods, proper = TRUE, print.flag = FALSE)
R> f15 <- glm.synds(smoke ~ sex  + edu  + sex * edu, data = s15,
+    family = "binomial")
R> compare(f15, 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 = s15)


Differences between results based on synthetic and observed data:
                  Std. coef diff p value CI overlap
(Intercept)               1.1613   0.412     0.7037
sexFEMALE                -0.5737   0.685     0.8536
eduVOC                   -0.5292   0.708     0.8650
eduSEC                   -0.4848   0.732     0.8763
eduHIGH                  -1.1151   0.430     0.7155
sexFEMALE:eduVOC         -0.4331   0.759     0.8895
sexFEMALE:eduSEC          1.0813   0.444     0.7241
sexFEMALE:eduHIGH         0.4852   0.732     0.8762


Measures for one synthesis and 8 coefficients
Mean confidence interval overlap:  0.813
Mean absolute std. coef diff:  0.733
Lack-of-fit: 4.497; p-value 0.81 for test that synthesis model is compatible
with a chi-squared test with 8 degrees of freedom


Confidence interval plot:

R> compare(f15, 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 = s15)


Differences between results based on synthetic and observed data:
                  Std. coef diff p value CI overlap
(Intercept)               1.1613   0.412     0.7887
sexFEMALE                -0.5737   0.685     0.7887
eduVOC                   -0.5292   0.708     0.7887
eduSEC                   -0.4848   0.732     0.7887
eduHIGH                  -1.1151   0.430     0.7887
sexFEMALE:eduVOC         -0.4331   0.759     0.7887
sexFEMALE:eduSEC          1.0813   0.444     0.7887
sexFEMALE:eduHIGH         0.4852   0.732     0.7887


Measures for one synthesis and 8 coefficients
```

```
Mean confidence interval overlap:  0.7887
Mean absolute std. coef diff:  0.733
Lack-of-fit: 4.497; p-value 0.81 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.