RESIDE

Table of contents

Authors

- Ryan Field
- Claudia Geue
- Neil Hawkins
- David McAllister
- Olivia Wu

Figure 1: University of Glasgow

An R package for communication about data held in safe havens

https://cran.r-project.org/web/packages/RESIDE/index.html

https://github.com/hehta/RESIDE

This work was supported by the UKRI Strength in Places Fund (SIPF) Competition, project number 107140. The project title is SIPF The Living Laboratory driving economic growth in Glasgow through real world implementation of precision medicine.

Principles of RESIDE

1. Easy for safe havens/trusted research environments
2. Transparently non-disclosive
3. Easy for end-users

Openness

- GPL (>= 3)
- Welcome to contribute (contributor statement coming soon)

Overview of data-flow

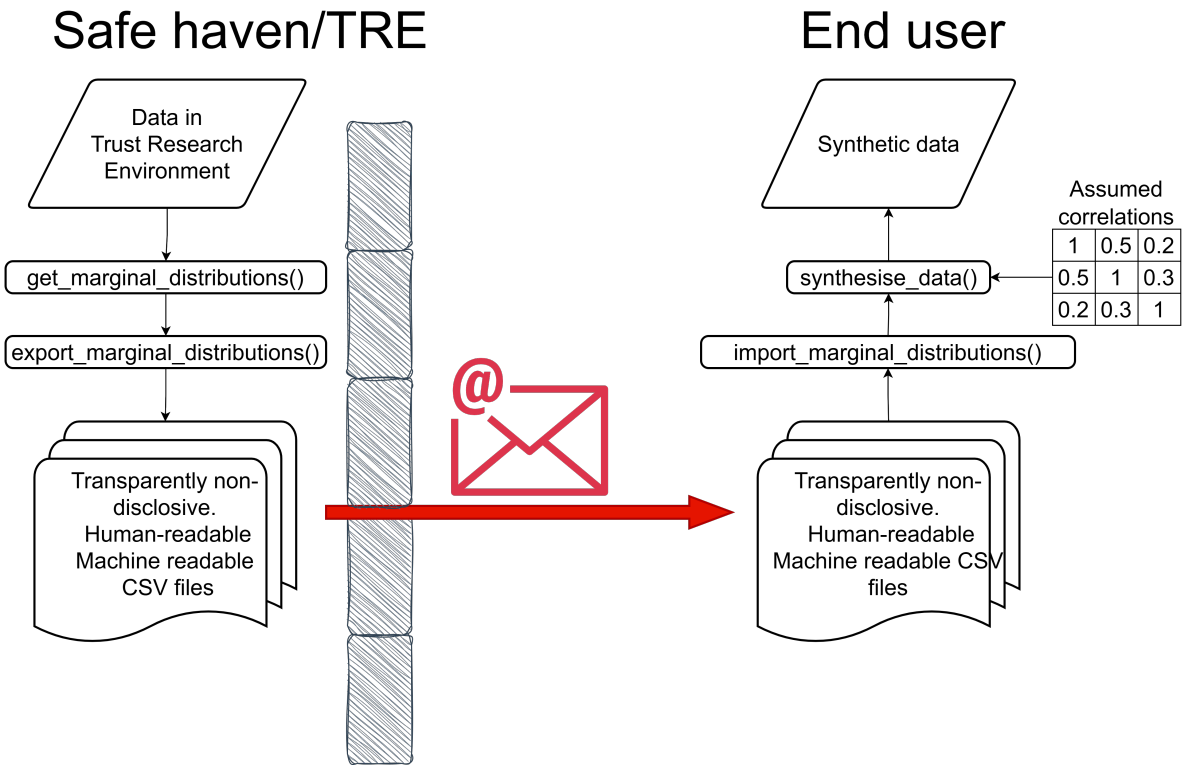


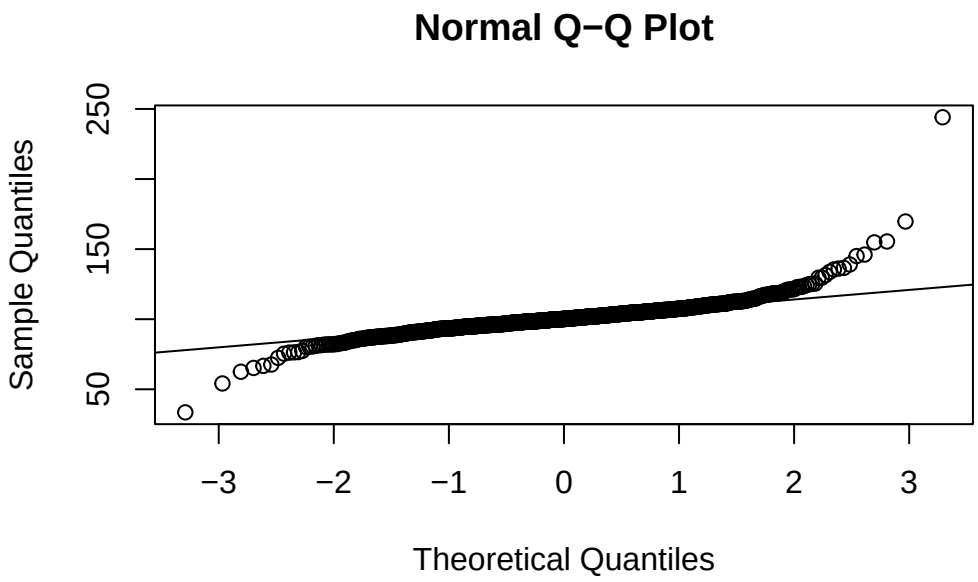
Figure 2: RESIDE overview

1. Easy to run for safe havens

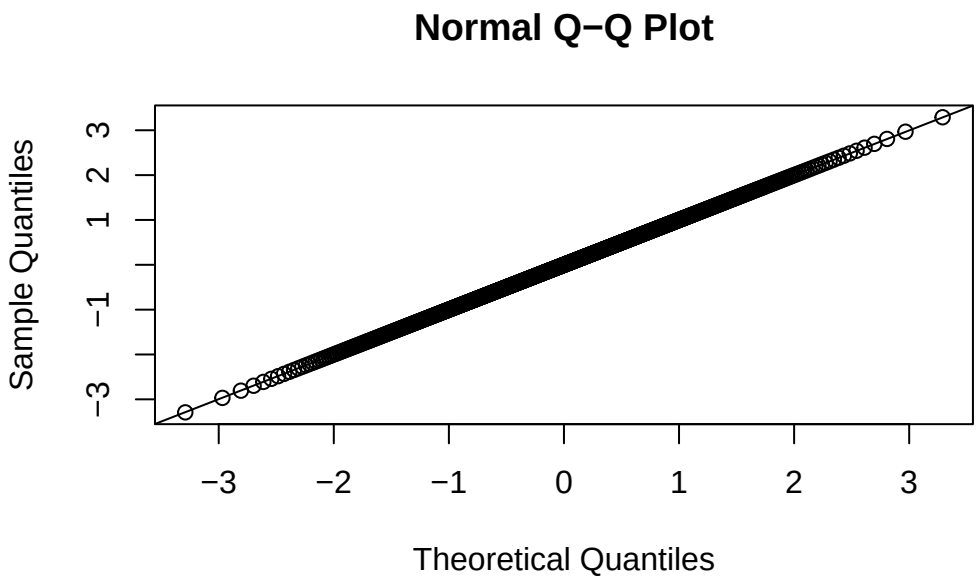
Ordered Quantile normalizing transformation

Some distribution

```
library(tidyverse)
some_var <- sort(100 + rt(1000, 3) * 6)
qqnorm(some_var)
qqline(some_var)
```

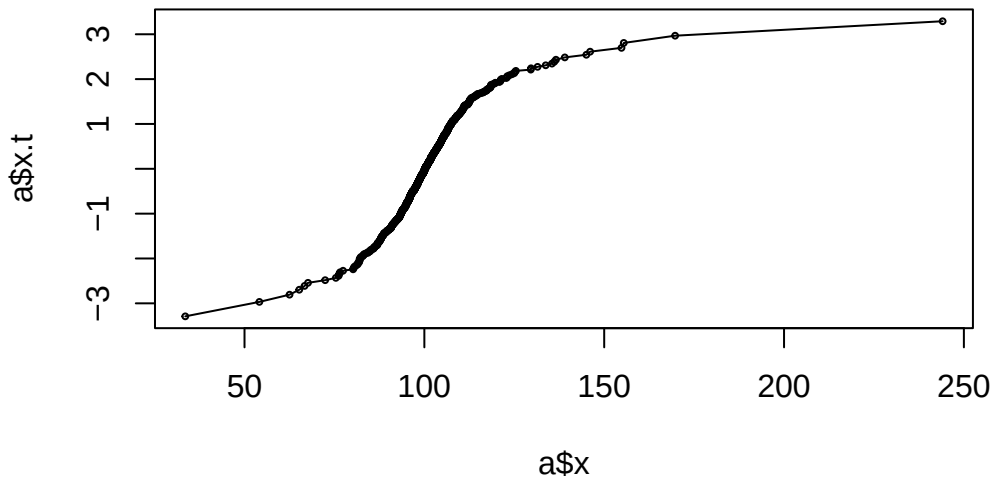



```
a <- bestNormalize::orderNorm(some_var)
a <- tibble(x = a$x,
            x.t = a$x.t) %>%
  distinct()
qqnorm(a$x.t)
qqline(a$x.t)
```



You can convert back and forward between the original/transformed using a look-up table of the data points, with linear interpolation

```
plot(a$x, a$x.t, pch = 1, cex = 0.4)
points(a$x, a$x.t, type = "l")
```



Concise

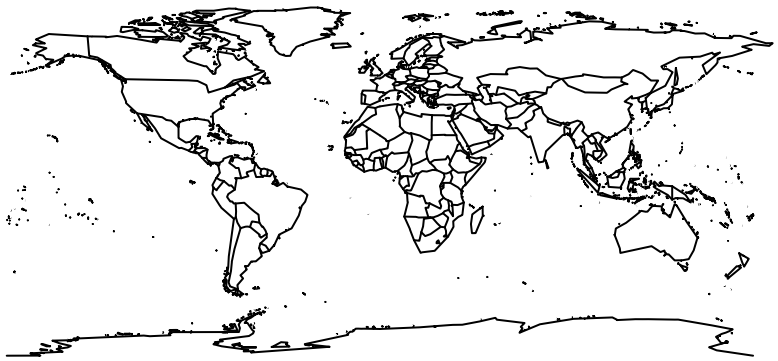
But safe havens wont want to export lots of points for every variable

```
a %>%
  distinct()
```

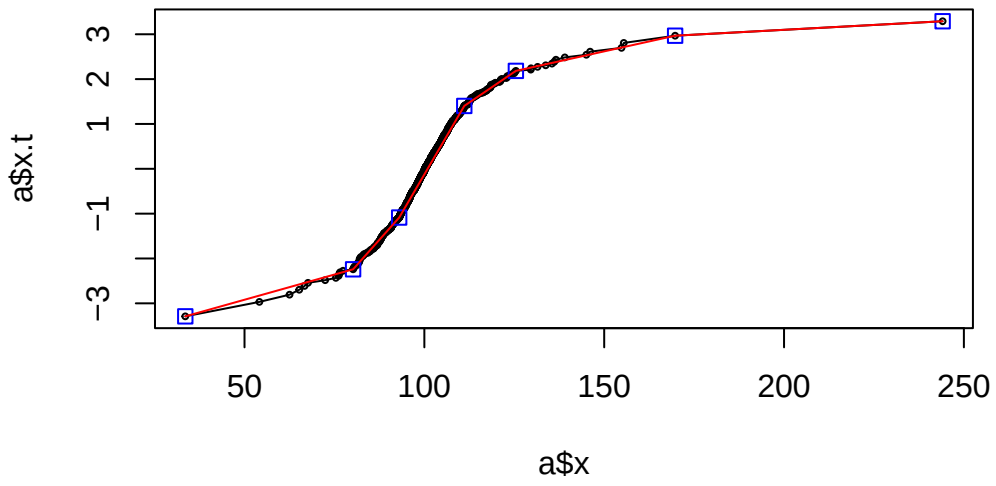
```
# A tibble: 1,000 x 2
       x   x.t
   <dbl> <dbl>
 1  33.5 -3.29
 2  54.1 -2.97
 3  62.5 -2.81
 4  65.2 -2.70
 5  66.7 -2.61
 6  67.7 -2.54
 7  72.4 -2.48
 8  75.4 -2.43
 9  76.2 -2.39
10  76.3 -2.35
# i 990 more rows
```

Ramer-Douglas-Peuker algorithm

```
maps::map("world")
```



```
res <- RDP::RamerDouglasPeucker(a$x, a$x.t, epsilon = 0.2)
res <- as_tibble(res) %>%
  rename(x.t = y)
plot(a$x, a$x.t, pch = 1, cex = 0.4)
points(a$x, a$x.t, type = "l")
points(res$x, res$x.t, pch = 0, col = "blue")
points(res$x, res$x.t, type = "l", col = "red")
```




```
res
```

```
# A tibble: 7 x 2
      x   x.t
  <dbl> <dbl>
1  33.5 -3.29
2  80.2 -2.24
3  93.0 -1.09
4 111.   1.40
5 125.   2.18
6 170.   2.97
7 244.   3.29
```

Summary

- No need to know/fit parametric distribution
- Small numbers of points to summarise distribution

Experience of running it for data holder/safe haven

Setup

```
# Load the Library
library(RESIDE)
```

Summarise Original Data

Select the variables of interest from the IST dataset and summarise:

```
# Select variables of interest from the IST dataset
IST_original <- IST |> dplyr::select(
  AGE,     # AGE at Randomisation
  SEX,     # SEX M/F
  RATRIAL, # Atrial Fibrillation Y/N at Randomisation
           # (not coded for 984 patients in the pilot phase)
  RSBP,    # Systolic Blood Pressure at Randomisation
  STRK14,  # Indicator of Any Stroke at 14 days,
  RDELAY   # Delay between stroke and randomisation in hours
)
```

Prepare data

Run a model on the original data

```
# Fit a model
res <- glm(STRK14 ~ AGE + SEX + RSBP + I(RDELAY/10),
                data = IST_original,
           family = "binomial")
```

Safe haven obtains the marginal distributions

Use `get_marginal_distributions()` to extract marginals for selected variables:

```r
marginals <- get_marginal_distributions(
  IST,
  variables = c(
    "AGE",
    "SEX",
    "RATRIAL",
    "RSBP",
    "STRK14",
    "RDELAY"
  )
)
```

Export the Marginal Distributions to text files

```r
if(!dir.exists("rhta_demo")) dir.create("rhta_demo")
export_marginal_distributions(
  marginals,
  folder_path = "rhta_demo",
  force = TRUE
)
```

2. Transparently non-disclosive

Open csv files and look at them

3. Experience of running it for data users

Received the marginals (download or email)

```r
imported_marginals <- import_marginal_distributions(
  folder_path = "rhta_demo"
)
```

Synthesise Data from Marginals (initially assuming independence)

- simstudy package
- extensively documented
- reasonably active development
- implemented sampling methods for different data types

```
sim_df <- synthesise_data(imported_marginals)
sim_df
```

```
           id SEX RATRIAL STRK14 AGE RSBP RDELAY
    1:      1   M       N      0  72  205     11
    2:      2   M       N      0  70  167     35
    3:      3   F       N      0  61  173     28
    4:      4   M       N      0  86  148     25
    5:      5   M       N      0  60  147     21
   ---
19431: 19431   F       N      0  77  163     13
19432: 19432   F       Y      0  63  243     14
19433: 19433   F       N      0  53  166     24
19434: 19434   M       N      0  79  120      8
19435: 19435   F       N      0  76  151     13
```

Fit a model using the Simulated Data

```
sim_df <- sim_df |>
  dplyr::mutate_if(is.character, factor)

# Fit the Cox PH model on the simulated data
simres1 <- glm(STRK14 ~ AGE + SEX + RSBP + I(RDELAY/10),
  data = sim_df,
  family = "binomial")
tibble(terms = names(coef(res)), real = coef(res) %>% round(3), sim1 = coef(simres1) %>% rou
```

```
# A tibble: 5 x 3
  terms          real   sim1
  <chr>         <dbl>  <dbl>
1 (Intercept)  -3.59  -3.50
2 AGE           0.007 -0.002
```

```
3 SEXM           0.045  0.135
4 RSBP           0.001  0.002
5 I(RDELAY/10) -0.103  0.027
```

Synthesise Data with Correlations

- As above, but specify correlations

1. **Export an empty correlation matrix**
2. **Edit it externally (e.g., Excel)**
3. **Reimport and specify correlations**
4. **Run `synthesise_data()` with the correlation matrix**

```r
# 1. Create an empty correlation matrix
export_empty_cor_matrix(
  imported_marginals,
  folder_path = "mycor"
)

# 2. Reimport the matrix
cor_matrix <- import_cor_matrix(
  file.path("mycor", "correlation_matrix.csv")
)
cor_matrix
```

```
                SEX_F SEX_M RATRIAL_missing RATRIAL_N RATRIAL_Y STRK14 AGE RSBP
SEX_F               1     0                0         0         0      0   0    0
SEX_M               0     1                0         0         0      0   0    0
RATRIAL_missing     0     0                1         0         0      0   0    0
RATRIAL_N           0     0                0         1         0      0   0    0
RATRIAL_Y           0     0                0         0         1      0   0    0
STRK14              0     0                0         0         0      1   0    0
AGE                 0     0                0         0         0      0   1    0
RSBP                0     0                0         0         0      0   0    1
RDELAY              0     0                0         0         0      0   0    0
                RDELAY
SEX_F                0
SEX_M                0
RATRIAL_missing      0
```

```
RATRIAL_N            0
RATRIAL_Y            0
STRK14               0
AGE                  0
RSBP                 0
RDELAY               1
```

```r
# 3. Add assumed correlations (symmetrically)
cor_matrix["RDELAY", "STRK14"] <- -0.05
cor_matrix["STRK14",    "RDELAY"] <- -0.05
set.seed(1234)
# 4. Synthesise data specifying the correlation matrix
sim_df_cor <- synthesise_data(
  imported_marginals,
  correlation_matrix = cor_matrix
)

sim_df_cor
```

```
          id SEX RATRIAL STRK14 AGE RSBP RDELAY
    1:     1   M       N      0  76  163     12
    2:     2   M       N      0  72  140     25
    3:     3   M       N      0  64  161     12
    4:     4   M       N      0  68  160     35
    5:     5   M       N      0  69  151     10
   ---
19431: 19431   M       N      0  86  192     21
19432: 19432   M       N      0  80  188      9
19433: 19433   M       N      0  73  199     25
19434: 19434   F       N      0  80  176      9
19435: 19435   M       N      0  83  159     18
```

Fit a model using the Simulated Data (With Correlations)

```r
# Fit the model on the synthesised data (with correlations)
simres2 <- glm(STRK14 ~ AGE + SEX + RSBP + I(RDELAY/10),
  data = sim_df_cor,
  family = "binomial")
tibble(terms = names(coef(res)),
```

```
        real = coef(res) %>% round(3),
        sim1 = coef(simres1) %>% round(3),
        sim2 = coef(simres2) %>% round(3))
```

```
# A tibble: 5 x 4
  terms          real   sim1   sim2
  <chr>         <dbl>  <dbl>  <dbl>
1 (Intercept)  -3.59  -3.50  -3.20
2 AGE           0.007 -0.002 -0.001
3 SEXM          0.045  0.135  0.126
4 RSBP          0.001  0.002  0.001
5 I(RDELAY/10) -0.103  0.027 -0.072
```

Summary

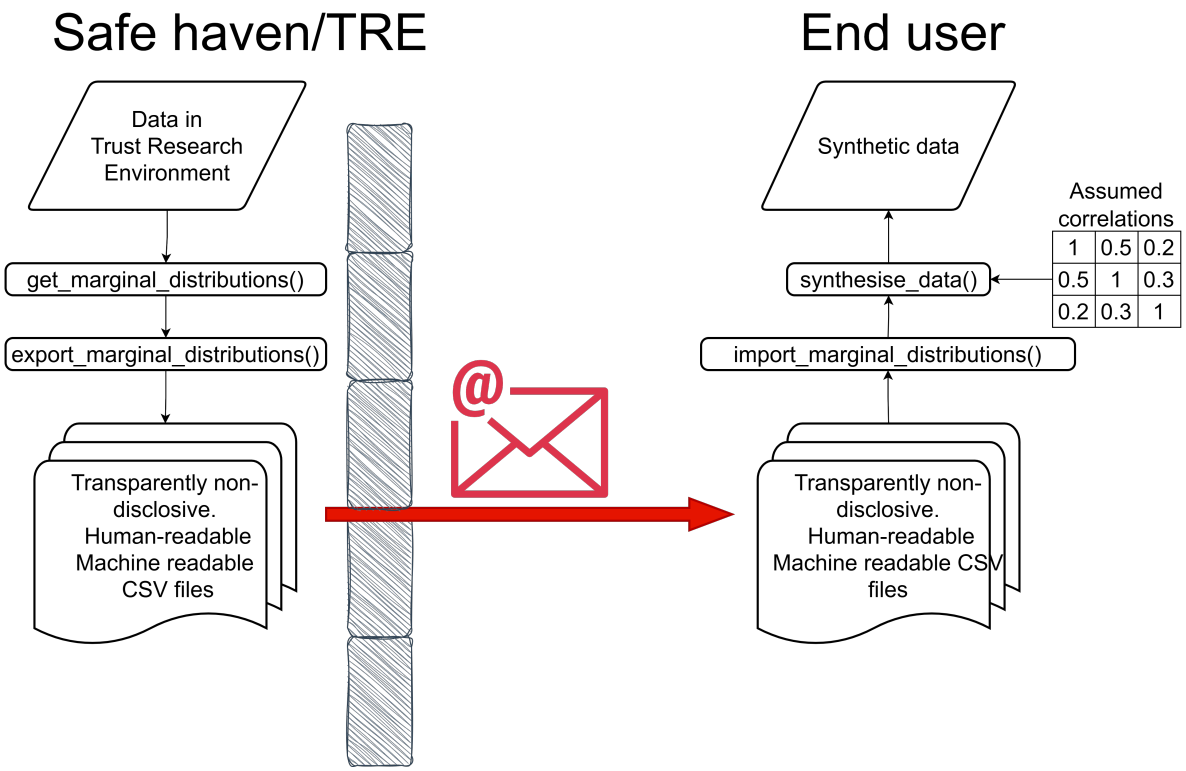


Figure 3: RESIDE overview

Limitations and next steps

- Multi-tables
- Categorical variables with lots of levels
  - Terminologies (eg ICD10/ICD11, WHO ATC)
- Categorical variables with >= 3 levels
  - Assume independent
  - Assume ordered
  - Allow nonsense combinations (eg blue eyes = 1, brown eyes = 1)