Dataset comparison using Synthpop

Intro

Synthpop (https://www.synthpop.org.uk/) is a package for the R programming language that us generally used to produce synthetic data for individual-level data. The synthpop package also provides a set of utilities and tools that allow you to compare two arbitrary datasets. These tools are generally used to compare original datasets with the results of creating synthetic data using synthpop, but we will be using them to compare original datasets with new synthetic datasets created using our own algorithms.

Installation

In order to use synthpop, you must first install the "R" programming language and its synthpop package. You don't need a working knowledge of R, as this guide will walk you through the process step by step, but you do need to have it installed. (If you already have R installed, skip to step "2" below.) **If you need help installing R or the synthpop library, please reach out to maia.hansen@gmail.com and schedule time to work through the process together.**

1. Install R using the instructions at https://cran.r-project.org/ If you are using Mac or Windows, you will likely want to use one of the pre-compiled binary packages available in the "Download and Install R" section. (If you are using Linux, you may want to use your Linux package manager instead.)
    a. **OPTIONAL**: Once you have installed R, you can optionally install the RStudio IDE at https://www.rstudio.com/products/rstudio/download/ . This is **not required** in order to run the synthpop package, but can be useful if you intend to do additional things with R.
2. Install the "devtools" R package.
    a. Open an R console by opening a terminal or command-line window and typing the single letter `r`
    b. Install the R "devtools" package with the following command:
    ```
    install.packages("devtools")
    ```
    c. You will be prompted to select a mirror site near you. R will provide you with a list of available mirror sites. Enter the number of the mirror site that is physically located nearest to you (e.g. "81" for Oregon, "17" for Beijing, etc.)
    d. You should see that R will automatically download and install the devtools package as well as any required dependencies.[1]
3. Load the "devtools" R package into your R terminal session with the following command:
    ```
    library(devtools)
    ```
    You should see the following message:
    ```
    > library(devtools)
    Loading required package: usethis
    ```
4. Next, install at least **version 1.7-0** of the synthpop package. (You will need version 1.7 in order to use the compare function with data that was not generated directly from synthpop.) Assuming you are connected to the internet, you can download this directly from github[2] with the following command:
    ```
    install_github("bnowok/synthpop")
    ```
    You should see the following message:

```
> install_github("bnowok/synthpop")
Downloading GitHub repo bnowok/synthpop@HEAD…
```

and the synthpop package will be installed.

If you did not receive any error messages during this process[1], you have successfully installed the required synthpop library! Continue to the next step, "Comparing Datasets."

Basic Dataset Comparison

This section assumes that you have two CSV (comma-separated values) files accessible locally on your filesystem. One should contain a ground truth dataset, and one should contain a dataset synthesized from this ground truth dataset.

This Quick Start guide will focus on performing the comparison using the R command line. For convenience, you may wish to write an R script that will perform the comparison for you, or use RStudio if you are already comfortable with R. That is outside the scope of this guide, but if you'd like to try that and want help, please feel free to contact maia.hansen@gmail.com.

To compare two datasets:

1. If you are not already in an R console, open one by opening a terminal or command-line window and typing the single letter `r` (followed by <ENTER>)
2. Load the synthpop library into your workspace with the following command:
   ```
   library(synthpop)
   ```
3. Load your ground truth dataset into an R dataframe with the following command:
   ```
   groundTruth <- read.csv(file = '/path/to/ground_truth.csv')
   ```
4. Load your synthetic dataset into an R dataframe with the following command:
   ```
   syntheticData <- read.csv(file = '/path/to/synthetic_data.csv')
   ```
5. Run a basic comparison between the two datasets with the following command:
   ```
   compare(syntheticData, groundTruth)
   ```

This comparison will open up a series of graphs that provide a basic comparison between the two datasets. Keep reading for examples and more tools!

Example

As an example, we will use the `ground_truth.csv` Chicago taxi driver dataset from Sprint 3. This dataset contains the following columns:

```
taxi_id,shift,company_id,pickup_community_area,dropoff_community_area,payment_type,trip_day_of_week,trip_hour_of_day,fare,tips,trip_total,trip_seconds,trip_miles
```

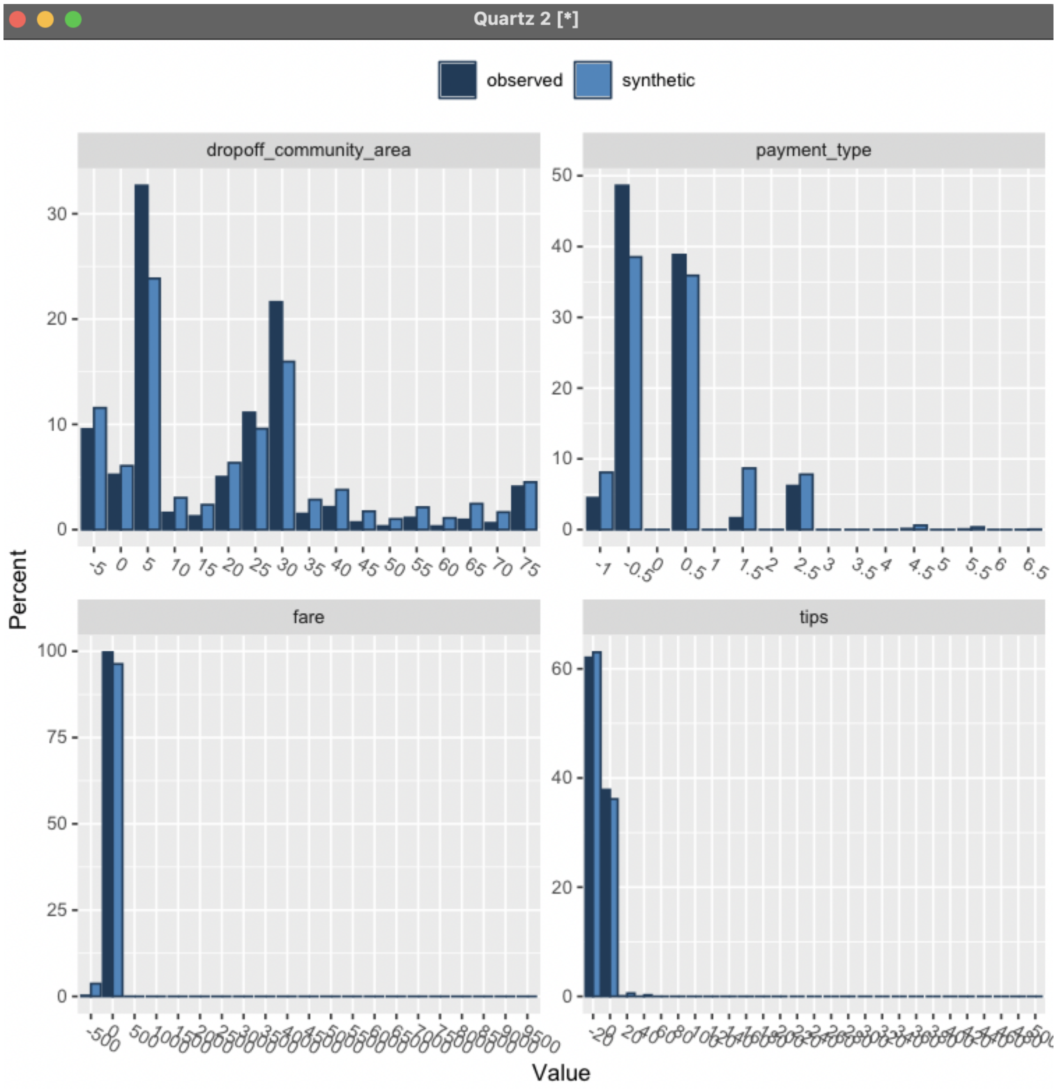
For our comparison dataset, we will use a synthetic dataset (named `submission.csv`), created as per the requirements of the Sprint 3 competition, containing the following columns:

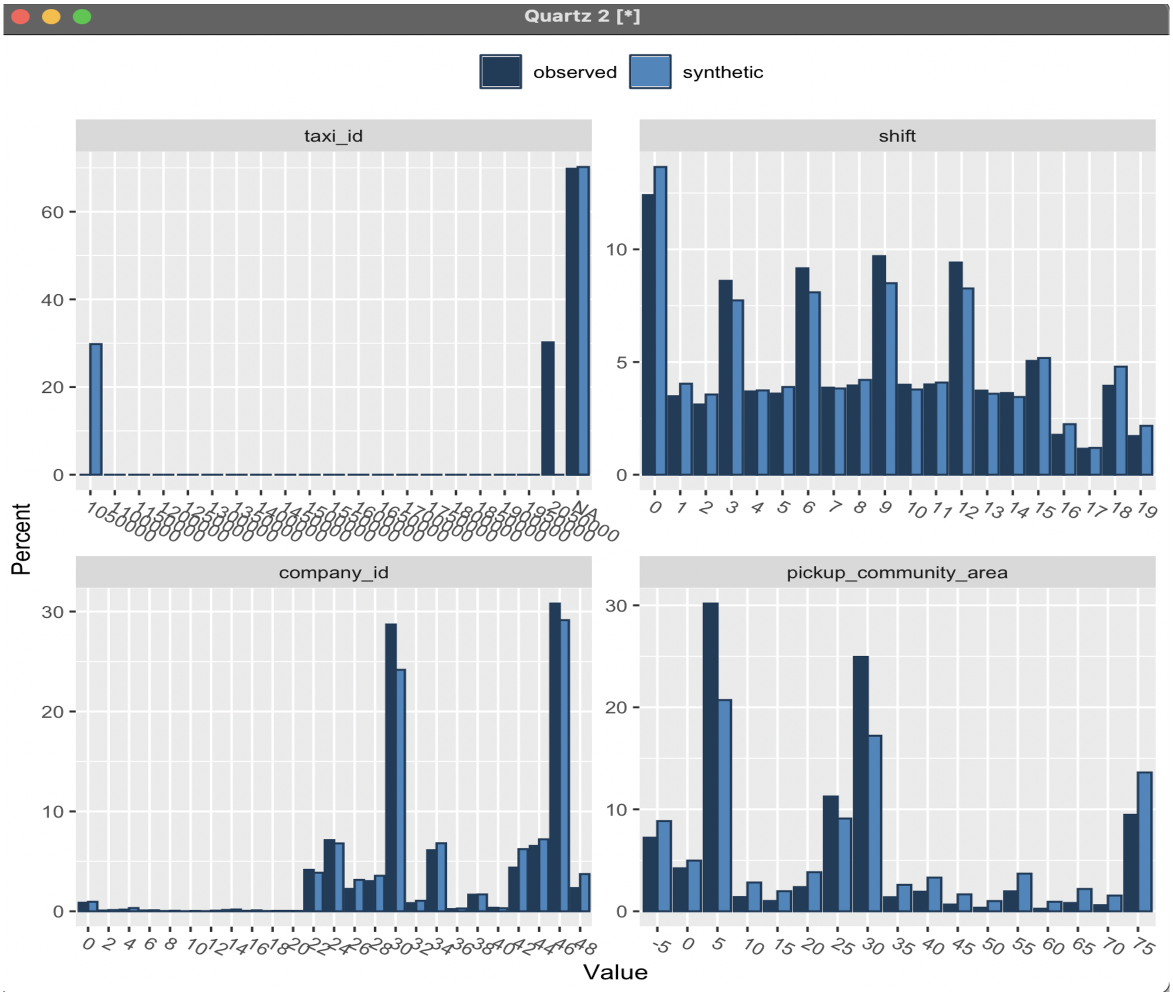
```
epsilon,taxi_id,shift,company_id,pickup_community_area,dropoff_community_area,payment_type,fare,tips,trip_total,trip_seconds,trip_miles
```

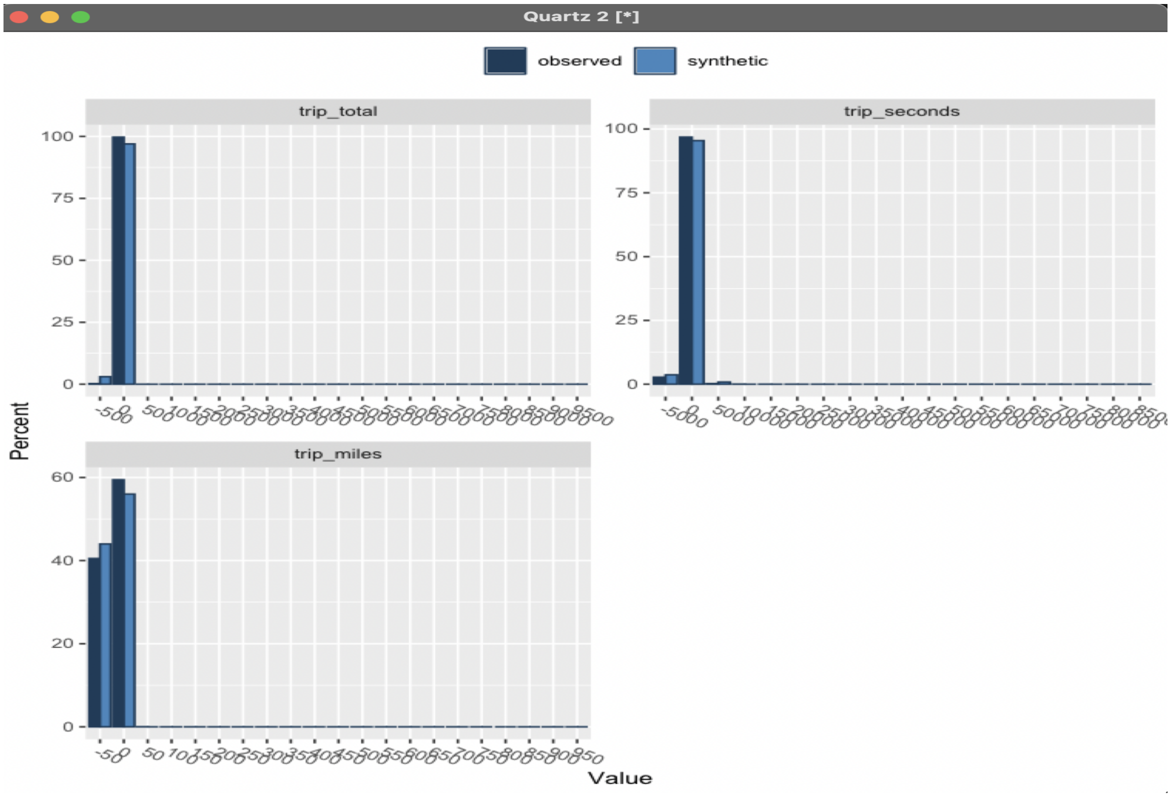
Both the `ground_truth.csv` and `submission.csv` files in this example are located in the current working directory. First, we enter an interactive R session by typing the letter `r` and hitting <ENTER>. Then, here is the R session that is run to compare these datasets, based on the instructions above[3]:

```
> library(synthpop)
Find out more at https://www.synthpop.org.uk/
> groundTruth <- read.csv(file = 'data/ground_truth.csv')
> syntheticData <- read.csv(file = 'submission.csv')
> compare(syntheticData, groundTruth)
Warning: Variable(s) epsilon in synthetic object but not in observed
data
  Looks like you might not have the correct data for comparison

Comparing percentages observed with synthetic

Press return for next variable(s):
Press return for next variable(s):
>
```

When the analysis is complete, graphs such as these will be displayed, comparing these two datasets:




Quartz 2 [*]
observed
synthetic
taxi_id
shift
company_id
pickup_community_area
Percent
Value
0
20
40
60
NA
0
5
10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
0
10
20
30
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48
0
10
20
30
-5 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75


Quartz 2 [*]
observed
synthetic
trip_total
trip_seconds
trip_miles
Percent
Value
0
25
50
75
100
0
25
50
75
100
0
20
40
60

Other Features

Synthpop also provides methods to perform other comparisons. Some features that might be useful include:

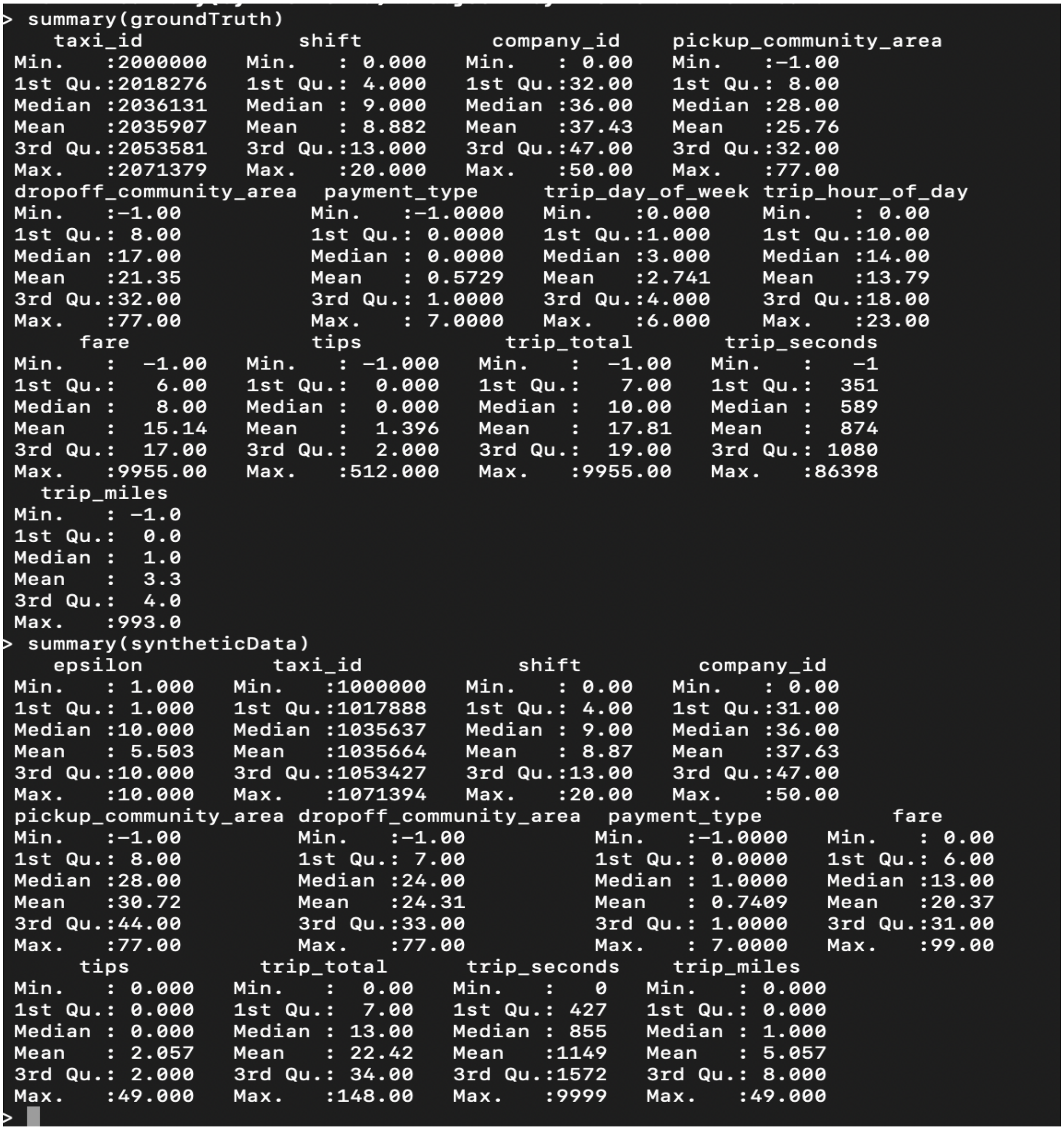
Dataset Summaries

To get a summary (min, median, max, etc.) of the values contained in a dataset, type the following command (using the examples loaded as described above):

```
summary(groundTruth)
```

Or

```
summary(syntheticData)
```



```
> summary(groundTruth)
    taxi_id            shift          company_id    pickup_community_area
 Min.   :2000000   Min.   : 0.000   Min.   : 0.00   Min.   :-1.00        
 1st Qu.:2018276   1st Qu.: 4.000   1st Qu.:32.00   1st Qu.: 8.00        
 Median :2036131   Median : 9.000   Median :36.00   Median :28.00        
 Mean   :2035907   Mean   : 8.882   Mean   :37.43   Mean   :25.76        
 3rd Qu.:2053581   3rd Qu.:13.000   3rd Qu.:47.00   3rd Qu.:32.00        
 Max.   :2071379   Max.   :20.000   Max.   :50.00   Max.   :77.00        
 dropoff_community_area  payment_type     trip_day_of_week trip_hour_of_day
 Min.   :-1.00          Min.   :-1.0000   Min.   :0.000    Min.   : 0.00   
 1st Qu.: 8.00          1st Qu.: 0.0000   1st Qu.:1.000    1st Qu.:10.00   
 Median :17.00          Median : 0.0000   Median :3.000    Median :14.00   
 Mean   :21.35          Mean   : 0.5729   Mean   :2.741    Mean   :13.79   
 3rd Qu.:32.00          3rd Qu.: 1.0000   3rd Qu.:4.000    3rd Qu.:18.00   
 Max.   :77.00          Max.   : 7.0000   Max.   :6.000    Max.   :23.00   
      fare               tips            trip_total        trip_seconds  
 Min.   :  -1.00   Min.   : -1.000   Min.   :  -1.00   Min.   :   -1  
 1st Qu.:   6.00   1st Qu.:  0.000   1st Qu.:   7.00   1st Qu.:  351  
 Median :   8.00   Median :  0.000   Median :  10.00   Median :  589  
 Mean   :  15.14   Mean   :  1.396   Mean   :  17.81   Mean   :  874  
 3rd Qu.:  17.00   3rd Qu.:  2.000   3rd Qu.:  19.00   3rd Qu.: 1080  
 Max.   :9955.00   Max.   :512.000   Max.   :9955.00   Max.   :86398  
   trip_miles   
 Min.   : -1.0  
 1st Qu.:  0.0  
 Median :  1.0  
 Mean   :  3.3  
 3rd Qu.:  4.0  
 Max.   :993.0  
> summary(syntheticData)
    epsilon          taxi_id            shift         company_id   
 Min.   : 1.000   Min.   :1000000   Min.   : 0.00   Min.   : 0.00  
 1st Qu.: 1.000   1st Qu.:1017888   1st Qu.: 4.00   1st Qu.:31.00  
 Median :10.000   Median :1035637   Median : 9.00   Median :36.00  
 Mean   : 5.503   Mean   :1035664   Mean   : 8.87   Mean   :37.63  
 3rd Qu.:10.000   3rd Qu.:1053427   3rd Qu.:13.00   3rd Qu.:47.00  
 Max.   :10.000   Max.   :1071394   Max.   :20.00   Max.   :50.00  
 pickup_community_area dropoff_community_area  payment_type          fare      
 Min.   :-1.00         Min.   :-1.00          Min.   :-1.0000   Min.   : 0.00  
 1st Qu.: 8.00         1st Qu.: 7.00          1st Qu.: 0.0000   1st Qu.: 6.00  
 Median :28.00         Median :24.00          Median : 1.0000   Median :13.00  
 Mean   :30.72         Mean   :24.31          Mean   : 0.7409   Mean   :20.37  
 3rd Qu.:44.00         3rd Qu.:33.00          3rd Qu.: 1.0000   3rd Qu.:31.00  
 Max.   :77.00         Max.   :77.00          Max.   : 7.0000   Max.   :99.00  
      tips          trip_total      trip_seconds    trip_miles    
 Min.   : 0.000   Min.   :  0.00   Min.   :   0   Min.   : 0.000  
 1st Qu.: 0.000   1st Qu.:  7.00   1st Qu.: 427   1st Qu.: 0.000  
 Median : 0.000   Median : 13.00   Median : 855   Median : 1.000  
 Mean   : 2.057   Mean   : 22.42   Mean   :1149   Mean   : 5.057  
 3rd Qu.: 2.000   3rd Qu.: 34.00   3rd Qu.:1572   3rd Qu.: 8.000  
 Max.   :49.000   Max.   :148.00   Max.   :9999   Max.   :49.000  
> 
```

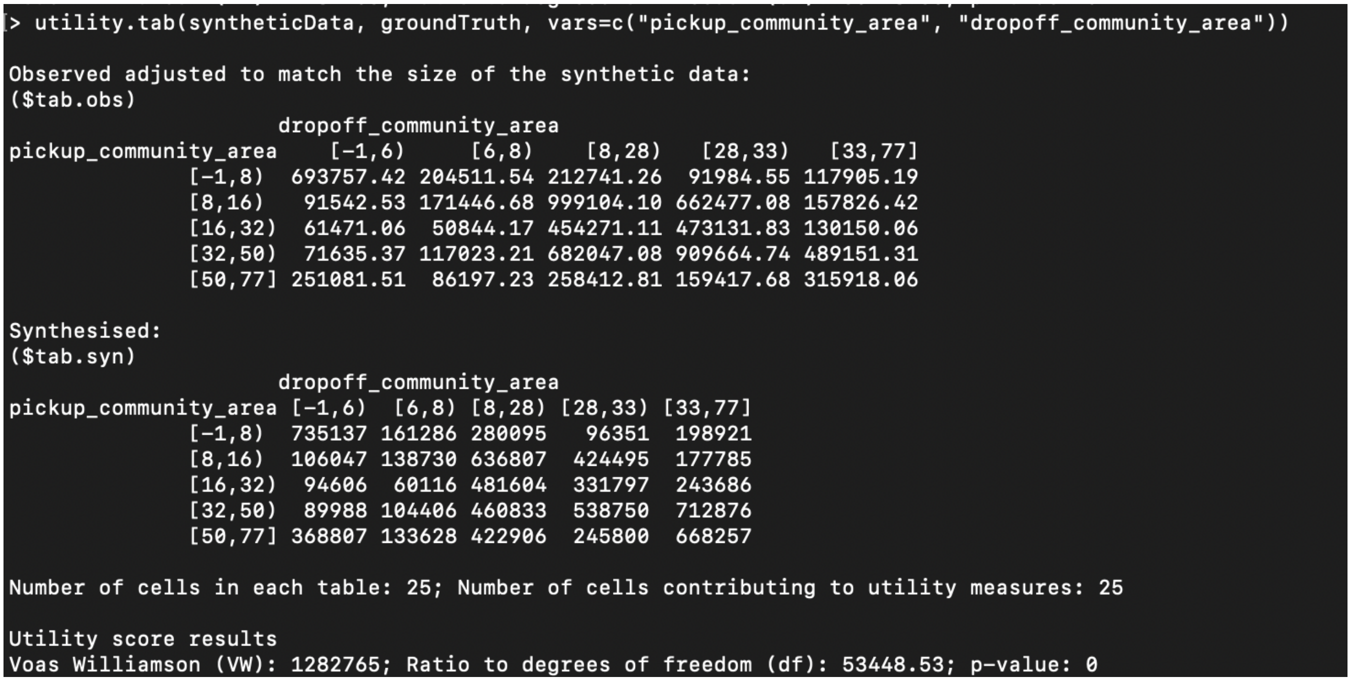
Tabular comparisons

Synthpop can also be used to produce and compare tables from observed and synthesised data. This uses the utility.tab function of synthpop, documented at https://github.com/bnowok/synthpop/blob/master/man/utility.tab.Rd

As an example, to produce a tabular comparison of the pickup and drop off community areas between the synthesized and ground truth datasets loaded in the example above, you would use the following command:

```
utility.tab(syntheticData, groundTruth, vars=c("pickup_community_area",
"dropoff_community_area"))
```

This produces results similar to the following:



```
[> utility.tab(syntheticData, groundTruth, vars=c("pickup_community_area", "dropoff_community_area"))

Observed adjusted to match the size of the synthetic data:
($tab.obs)
                     dropoff_community_area
pickup_community_area    [-1,6)     [6,8)    [8,28)   [28,33)   [33,77]
              [-1,8)  693757.42 204511.54 212741.26  91984.55 117905.19
              [8,16)   91542.53 171446.68 999104.10 662477.08 157826.42
              [16,32)  61471.06  50844.17 454271.11 473131.83 130150.06
              [32,50)  71635.37 117023.21 682047.08 909664.74 489151.31
              [50,77] 251081.51  86197.23 258412.81 159417.68 315918.06

Synthesised:
($tab.syn)
                     dropoff_community_area
pickup_community_area [-1,6)  [6,8) [8,28) [28,33) [33,77]
              [-1,8)  735137 161286 280095   96351  198921
              [8,16)  106047 138730 636807  424495  177785
              [16,32)  94606  60116 481604  331797  243686
              [32,50)  89988 104406 460833  538750  712876
              [50,77] 368807 133628 422906  245800  668257

Number of cells in each table: 25; Number of cells contributing to utility measures: 25

Utility score results
Voas Williamson (VW): 1282765; Ratio to degrees of freedom (df): 53448.53; p-value: 0
```

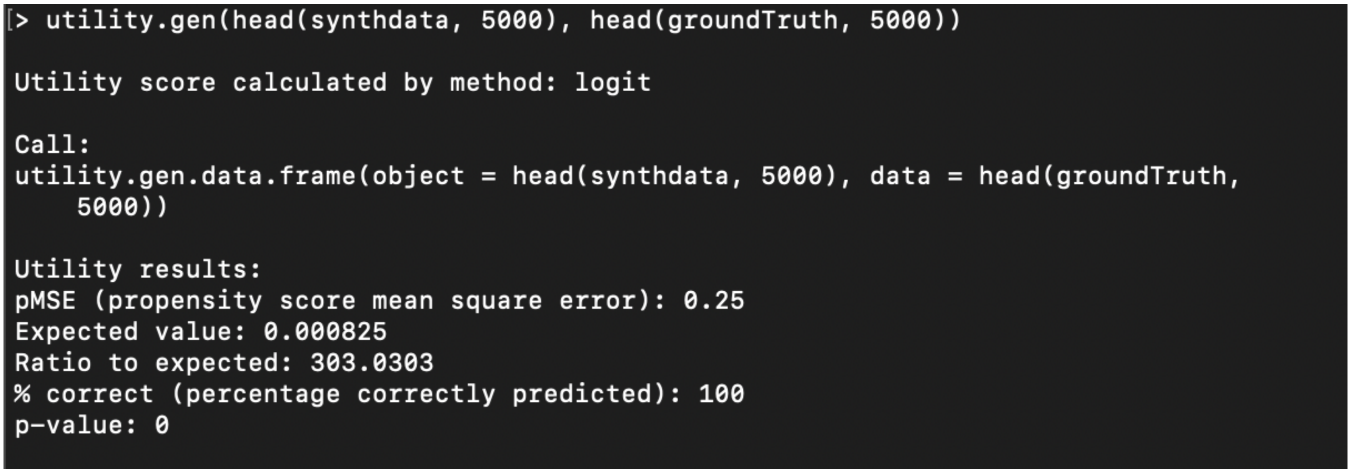
Distributional comparisons

You can also use synthpop to do a distributional comparison of synthesized and ground truth data, using the utility.gen function of synthpop, documented at https://github.com/bnowok/synthpop/blob/master/man/utility.gen.Rd. In order to use it, the synthetic data must not contain any columns that are not also present in the ground truth data. However, in our example above, the synthetic data contains the column `epsilon`, which is not present in the original data.

In order to delete the `epsilon` column from the data loaded as `syntheticData` in the example above, run the following command (after loading the data in your R console) to create a new `synthdata` dataframe that does not have `epsilon`:

```
    synthdata = subset(syntheticData, select = -c(epsilon))
```

You should now be able to run the `utility.gen` function to compare the two datasets. However, this function requires a great deal of internal memory, and so you may need to select only the first N rows of each dataset in order to get a result, using the `head` function. As an example, here is a comparison of the first 5000 rows of our ground truth and synthetic data (after removing the `epsilon` column), as described in the example above. (Note that this actual example score is not very good / valid, due to the arbitrary subsetting we did with `head`).



```
[> utility.gen(head(synthdata, 5000), head(groundTruth, 5000))

Utility score calculated by method: logit

Call:
utility.gen.data.frame(object = head(synthdata, 5000), data = head(groundTruth,
    5000))

Utility results:
pMSE (propensity score mean square error): 0.25
Expected value: 0.000825
Ratio to expected: 303.0303
% correct (percentage correctly predicted): 100
p-value: 0
```

Refer to the synthpop documentation for the `utility.gen` function located at https://github.com/bnowok/synthpop/blob/master/man/utility.gen.Rd#L120 for more details on how to use and configure the `utility.gen` function to produce more useful and interesting results.

For more details on synthpop, refer to its github repository located at https://github.com/bnowok/synthpop.

You can also access the full documentation for synthpop at https://rdrr.io/cran/synthpop/ -- be aware, though, that this documentation is for version 1.6.0, which does NOT support comparisons between arbitrary datasets. (It only supports comparisons between ground truth datasets and synthetic datasets created using synthpop itself.) However, it may still be useful for finding information about basic functionality.

Python

There does exist at least one Python implementation of synthpop, at https://hazy.com/blog/2020/01/31/synthpop-for-python. We have not tried this implementation, as the native R implementation has the latest functionality that supports direct comparisons between arbitrary datasets. However, if you'd like to try this out, please let us know how it goes!

Footnotes

1. If you are running this on a Mac, you may see a warning message similar to the following:

```
Warning message:
In doTryCatch(return(expr), name, parentenv, handler) :
  unable to load shared object
'/Library/Frameworks/R.framework/Resources/modules//R_X11.so':

dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11
.so, 6): Library not loaded: /opt/X11/lib/libSM.6.dylib
```

```
  Referenced from:
/Library/Frameworks/R.framework/Versions/4.1/Resources/modules/
R_X11.so
  Reason: image not found
```

This error message is benign and can be safely ignored.

2. If you are familiar with using R and the CRAN package manager, please note that the version of synthpop that is available directly from the R package manager has *not* yet been updated to 1.7, and so you will need to follow this process of installing from github.
3. Notice that when you run the compare function, you'll receive the following message:

```
     Warning: Variable(s) epsilon in synthetic object but not in
observed data
      Looks like you might not have the correct data for comparison
```

This is expected, because the "epsilon" column in the synthesized data does not exist in the ground truth data. The message can be ignored.

Background

This document was originally compiled Aug 13, 2021, by Maia Hansen (https://github.com/hd23408) for use in the NIST 2020 Differential Privacy Temporal Map Challenge.