Analysis

Yue Xiong

2022-09-15

Step1: Data Preparation

In step1, we aim to read in the original datasets and the synthetic dataset while aligning the variables especially set in the syn dataset. Attempting to compare the two datasets, we should also concatenate the original datasets vertically with variables all set as the same. Also, we are filtering out all the non-GPDR countries.

```
# set the working directory
wd <- getwd()
setwd(wd)
# read in the synthetic dataset
syn_data <- read.csv(file = "./syn_2020-08-02_2020-08-08.csv")
# and on macos
head(syn_data)
```

```
##   sample_weight B1_1 B1_2 B1_3 B1_4 B1_5 B1_6 B1_7 B1_8 B1_9 B1_10 B1_11 B1_12
## 1      20424.14    2    2    2    2    2    2    2    2    2     2     2     2
## 2      20424.14    2    2    2    1    2    2    2    2    2     2     2     2
## 3      20424.14    2    1    1    2    2    1    2    2    2     2     2     2
## 4      20424.14    2    2    2    2    2    2    2    2    2     2     2     2
## 5      20424.14    2    1    2    1    1    1    1    1    1     2     2     1
## 6      20424.14    2    2    2    1    2    2    2    2    2     2     2     2
##   B1_13 B1b_x1 B1b_x2 B1b_x3 B1b_x4 B1b_x5 B1b_x6 B1b_x7 B1b_x8 B1b_x9 B1b_x10
## 1     2    -99    -99    -99    -99    -99    -99    -99    -99    -99     -99
## 2     2    -99    -99    -99      1    -99    -99    -99    -99    -99     -99
## 3     2    -99      2      1    -99    -99      2    -99    -99    -99     -99
## 4     2    -99    -99    -99    -99    -99    -99    -99    -99    -99     -99
## 5     2    -99      2    -99      2      2      2      2      2      2     -99
## 6     2    -99    -99    -99      2    -99    -99    -99    -99    -99     -99
##   B1b_x11 B1b_x12 B1b_x13 B3  B5 B6  B7  B8  B9 B10 B11 B12_1 B12_2 B12_3 B12_4
## 1     -99     -99     -99  2 -99  2 -99 -99 -99 -99 -99   -99   -99   -99   -99
## 2     -99     -99     -99  2 -99  1   1   2   2 -99 -99   -99   -99   -99   -99
## 3     -99     -99     -99  2 -99  2 -99 -99 -99 -99 -99   -99   -99   -99   -99
## 4     -99     -99     -99  2 -99  2 -99 -99 -99 -99 -99   -99   -99   -99   -99
## 5     -99       2     -99  2 -99  2 -99 -99 -99 -99 -99   -99   -99   -99   -99
## 6     -99     -99     -99  2 -99  2 -99 -99 -99 -99 -99   -99   -99   -99   -99
##   B12_5 B12_6 B13_1 B13_2 B13_3 B13_4 B13_5 B13_6 B13_7 B14_1 B14_2 B14_3 B14_4
## 1   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
## 2   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
## 3   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
## 4   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
```

```
## 5   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
## 6   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99   -99
##   B14_5 C0_1 C0_2 C0_3 C0_4 C0_5 C0_6 C1_m  C2 C3 C5 C6 C7 C8 D1 D2 D3 D4 D5
## 1   -99    2    1    2    1    2    2    2 -99  2  5  3  4  1  5  5  3  4  3
## 2   -99    2    2    2    2    2    2    1   1  2  1  1  3  1  5  5  3  4  4
## 3   -99    2    1    2    2    2    2    1   2  2  1  1  4  1  5  4  3  3  3
## 4   -99    2    1    1    1    2    2    1   1  2  1  3  4  1  5  4  2  3  2
## 5   -99    2    1    2    2    2    1    2 -99  2  2  3  4  1  4  2  2  3  1
## 6   -99    2    1    2    1    2    1    1   2  2  2  3  4  1  5  4  3  4  4
##   D6_1 D6_2 D6_3 D7  D8  D9 D10 E2  E3 E4 E7 F1 F2_1 F2_2 F3_de GID_0    GID_1
## 1  -99  -99  -99  2   2 -99 -99  3   1  6  1  1    2    2   -99   NLD  NLD.8_1
## 2  -99  -99  -99  2   2 -99 -99  2   1  6  1  1    2    2   -99   FRA  FRA.7_1
## 3  -99  -99  -99  2   2 -99 -99  1   1  6  2  1    2    2   -99   ITA ITA.11_1
## 4  -99  -99  -99  1 -99 -99  13  2   1  5  3  1    2    2   -99   HUN HUN.14_1
## 5  -99  -99  -99  2   1   7   2  1   2  2  2  1    2    2   -99   FIN  FIN.5_1
## 6  -99  -99  -99  2   2 -99 -99  1 -99  1  5  1    2    2   -99   SVK  SVK.3_1
##           B2  B4        E5      E6
## 1        -99 -99    [2, 4)     -99
## 2   [28, 90) -99    [1, 2)  [0, 9)
## 3 [180, 366) -99    [2, 4) [9, 26)
## 4        -99 -99    [2, 4) [9, 26)
## 5     [1, 3) -99    [1, 2) [9, 26)
## 6        -99 -99 [6, 1000) [9, 26)
```

```
# syn_data <- read_csv("./syn_2020-08-02_2020-08-08.csv", show_col_types = FALSE)
# rename "sample weight" to "weight" to avoid conflicts
colnames(syn_data)[colnames(syn_data) == "sample_weight"] <- "weight"

# columns to be included from the original dataset
cols_list <- colnames(syn_data)

# initialize an empty dataset list
ori_dataset <- list()

for (i in 1:7){
ori_dataset[[i]] <- vroom(list.files(pattern = "*_full.csv$")[i],
                    show_col_types = FALSE) %>%
                    select(all_of(cols_list))
}

dim(ori_dataset[[2]])[2] == dim(syn_data)[2]
```

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

```
# check whether 2 dimensions coincide with each other

# bind the original datasets from 0802 to 0808 vertically
bindori_dataset <- bind_rows(ori_dataset)
bindori_dataset <- as.data.frame(bindori_dataset)
dim(bindori_dataset)
```

```
## [1] 996174     92
```

```
dim(syn_data)
```

```
## [1] 100000     92
```

Now we need to check all the `gpdr` countries and only do the alignment based on `gpdr` countries.

```
gpdr_countries_data <- NA
gpdr_countries_data <- read.csv(file = "./gpdr.csv", sep = ",")
head(gpdr_countries_data$Country_GID, n = 10L)
```

```
##  [1] "AUT" "AUT" "AUT" "AUT" "AUT" "AUT" "AUT" "AUT" "AUT" "BEL"
```

We then filter the binded original datasets and syn dataset with `Region_GID` specified

```
country_name <- unique(as.character(gpdr_countries_data$Country_GID))
length(country_name)
```

```
## [1] 27
```

```
length(unique(bindori_dataset$GID_0))
```

```
## [1] 218
```

```
bindori_dataset_filtered <- bindori_dataset %>%
                                        filter(GID_0 %in% country_name)


syn_data_filtered <- syn_data %>%
                           filter(GID_0 %in% country_name)
length(unique(bindori_dataset_filtered$GID_0)) == length(unique(syn_data_filtered$GID_0))  # check whet
```

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

Step2: make all the vars aligned with each other

These data that need such alignment are the first filtered ones, `syn_data_filtered` and `bindori_dataset_filtered`. They are filtered only with those gpdr countries filtered. First of all, we need to output the types of all the variables included in the two datasets and compare the difference at the first place.

Apparently shown above, one thing to find is that all the previously listed `double`-type variables, such as the `E4` variable. We should firstly collect all these variables and change the type of the original variables.

2.1 numeric to integer

```
vars_list <- NULL
int_vars_list <- c()

for (i in 1:ncol(syn_data_filtered)) {
  if(class(syn_data_filtered[[i]])=="integer") {
    # print(i)
    int_vars_list <- append(int_vars_list, colnames(syn_data_filtered[i]))
  }
}

length(int_vars_list)
```

```
## [1] 85
```

So let us convert all the numeric variables to integer.

```
bindori_dataset_filtered[int_vars_list] <- sapply(bindori_dataset_filtered[int_vars_list], as.integer)

# check whether all the previous numeric types are converted to integer
sapply(bindori_dataset_filtered[int_vars_list], class)
```

```
##      B1_1      B1_2      B1_3      B1_4      B1_5      B1_6      B1_7      B1_8
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##      B1_9     B1_10     B1_11     B1_12     B1_13    B1b_x1    B1b_x2    B1b_x3
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##    B1b_x4    B1b_x5    B1b_x6    B1b_x7    B1b_x8    B1b_x9   B1b_x10   B1b_x11
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##   B1b_x12   B1b_x13        B3        B5        B6        B7        B8        B9
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##       B10       B11     B12_1     B12_2     B12_3     B12_4     B12_5     B12_6
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##     B13_1     B13_2     B13_3     B13_4     B13_5     B13_6     B13_7     B14_1
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##     B14_2     B14_3     B14_4     B14_5      C0_1      C0_2      C0_3      C0_4
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##      C0_5      C0_6      C1_m        C2        C3        C5        C6        C7
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##        C8        D1        D2        D3        D4        D5      D6_1      D6_2
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##      D6_3        D7        D8        D9       D10        E2        E3        E4
## "integer" "integer" "integer" "integer" "integer" "integer" "integer" "integer"
##        E7        F1      F2_1      F2_2     F3_de
## "integer" "integer" "integer" "integer" "integer"
```

2.2 var B2

Firstly, we change the remaining “B2”, “B4”, “E5”, “E6” to character type in the original dataframe.

For variable B2, let’s take a look at the table of the occurrences of indicators.

And for the synthetic dataset, we observed that it is the “character” type.

In order to align with the synthetic dataset types, we make all the cell values corresponded to the the value/threshold shown above.

```
bindori_dataset_filtered["B2"][bindori_dataset_filtered["B2"] == -99] <- "-99"

bindori_dataset_filtered["B2"][bindori_dataset_filtered["B2"] >= 1000] <- "1000"

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 0 & bindori_dataset_filtered$B2 < 1] <- "[0,

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 1 & bindori_dataset_filtered$B2 < 3] <- "[1,

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 3 & bindori_dataset_filtered$B2 < 8] <- "[3,

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 8 & bindori_dataset_filtered$B2 < 15] <- "[8

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 15 & bindori_dataset_filtered$B2 < 28] <- "[

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 28 & bindori_dataset_filtered$B2 < 90] <- "[

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 90 & bindori_dataset_filtered$B2 < 180] <- "

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 180 & bindori_dataset_filtered$B2 < 366] <-

bindori_dataset_filtered$B2[bindori_dataset_filtered$B2 >= 366 & bindori_dataset_filtered$B2 < 1000] <-
```

```
# quickly check the class
class(bindori_dataset_filtered$B2)
```

```
## [1] "character"
```

3.3 var B4

```
bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 < 0] <- "-99"

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 1000] <- "1000"

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 0 & bindori_dataset_filtered$B4 < 1] <- "[0,

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 1 & bindori_dataset_filtered$B4 < 5] <- "[1,

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 5 & bindori_dataset_filtered$B4 < 10] <- "[5

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 10 & bindori_dataset_filtered$B4 < 1000] <-
```

```
# quickly check the class
class(bindori_dataset_filtered$B4)
```

```
## [1] "character"
```

3.4 var E5

```
bindori_dataset_filtered$E5[bindori_dataset_filtered$E5 < 0] <- "-99"

bindori_dataset_filtered$E5[bindori_dataset_filtered$E5 >= 1000] <- "1000"

bindori_dataset_filtered$E5[bindori_dataset_filtered$E5 >= 0 & bindori_dataset_filtered$E5 < 1] <- "[0,

bindori_dataset_filtered$E5[bindori_dataset_filtered$E5 >= 1 & bindori_dataset_filtered$E5 < 2] <- "[1,

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 2 & bindori_dataset_filtered$B4 < 4] <- "[2,

bindori_dataset_filtered$B4[bindori_dataset_filtered$B4 >= 4 & bindori_dataset_filtered$B4 < 6] <- "[4,

bindori_dataset_filtered$E5[bindori_dataset_filtered$E5 >= 6 & bindori_dataset_filtered$E5 < 1000] <- "
```

```
# quickly check the class
class(bindori_dataset_filtered$E5)
```

```
## [1] "character"
```

3.5 var E6

```
bindori_dataset_filtered$E6[bindori_dataset_filtered$E6 < 0] <- "-99"

bindori_dataset_filtered$E6[bindori_dataset_filtered$E6 >= 26] <- "26"

bindori_dataset_filtered$E6[bindori_dataset_filtered$E6 >= 0 & bindori_dataset_filtered$E6 < 9] <- "[0,

bindori_dataset_filtered$E6[bindori_dataset_filtered$E6 >= 9 & bindori_dataset_filtered$E6 < 26] <- "[9
```

```
# quickly check the class
class(bindori_dataset_filtered$E6)
```

```
## [1] "character"
```

3.6 check for observations for var D1 and D2(ori)

For D1, we have

```
table(bindori_dataset_filtered$D1)
```

```
##
##    -99      1      2      3      4      5
##  18545   2111   6152  26588  45796 161107
```

```
table(syn_data_filtered$D1)
```

```
##
##   -99     1     2     3     4     5
##  1043   857  2361 10605 18886 66248
```

For D2, we have

```
table(bindori_dataset_filtered$D2)
```

```
##
##    -99      1      2      3      4      5
##  16316   2687   7669  27035  46831 159761
```

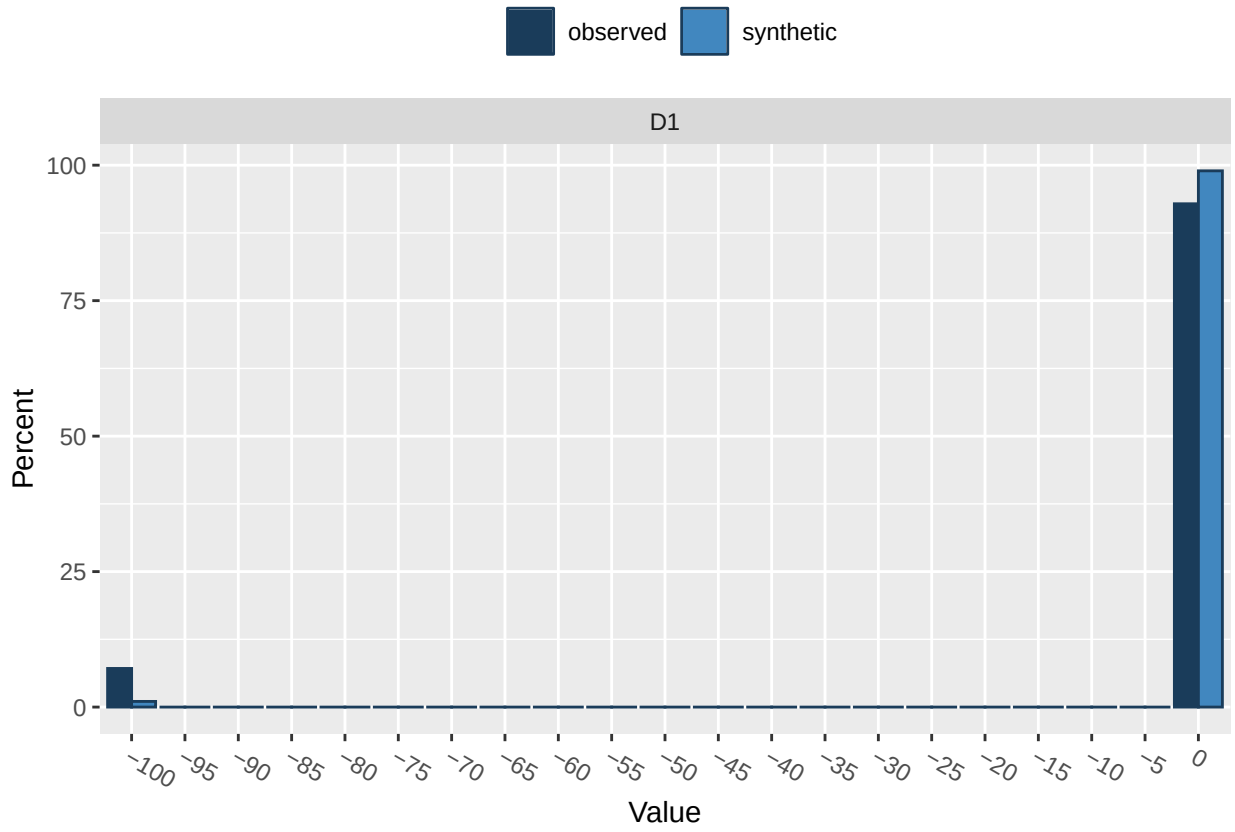
```
table(syn_data_filtered$D2)
```

```
##
##   -99     1     2     3     4     5
##   114  1138  3056 11256 19280 65156
```

3.7 re-evaluate with aligned variables

```
## for var D1 -> module B
compare(object = data.frame(D1 = syn_data_filtered$D1),
        data = data.frame(D1 = bindori_dataset_filtered$D1),
        vars = "D1", cont.na = NULL,
        msel = NULL, stat = "percents",
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

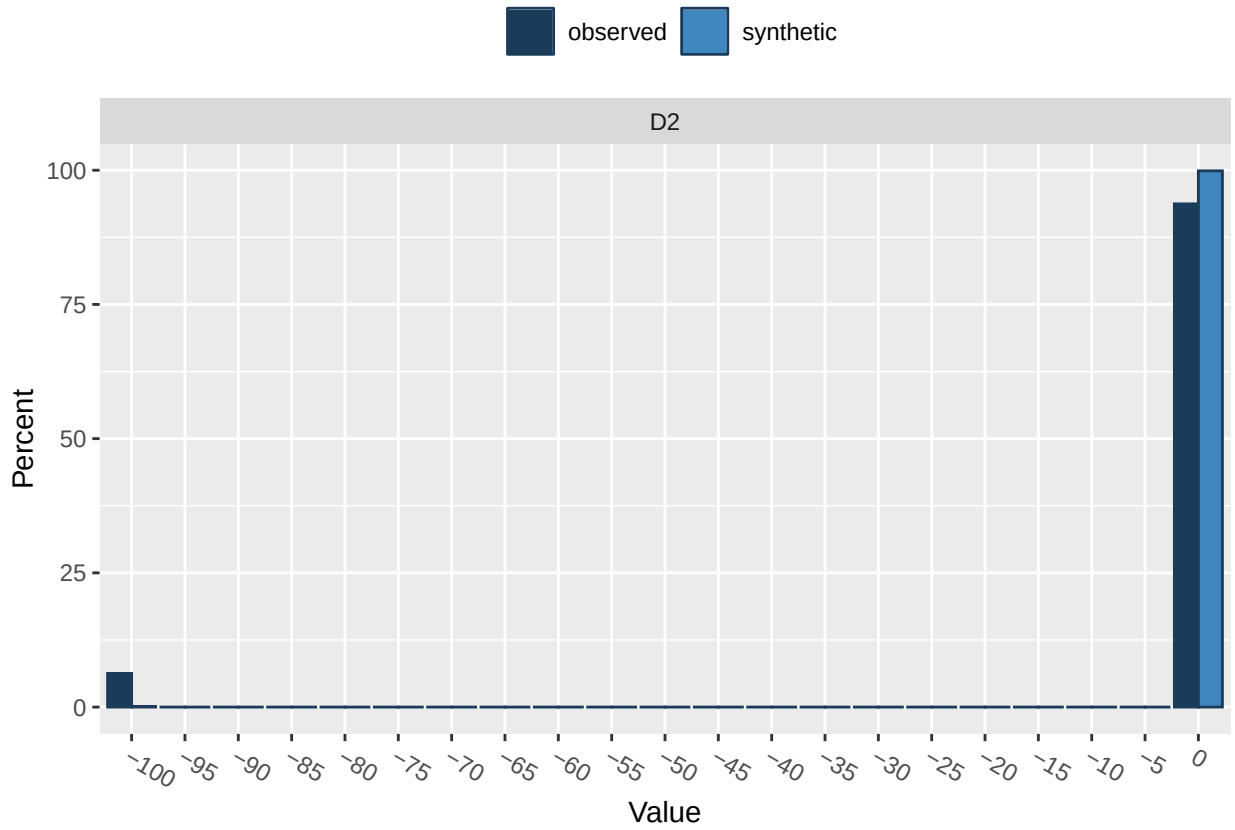
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## D1 0.000835 2076.806
```

```
## for var D2 -> module B
compare(object = data.frame(D2 = syn_data_filtered$D2),
        data = data.frame(D2 = bindori_dataset_filtered$D2),
        vars = "D2", cont.na = NULL,
        msel = NULL, stat = "percents",
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

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




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## D2 0.000665 1654.444
```

Let's first prepare the variables to be included in the comparison.

Step3: Evaluating the utility of the syn data

In step3, we try to evaluate the utility of the synthetic dataset with one-way marginal and two-way marginal measures.

- As for the one-way utility, the syn dataset is measured with the compare plots and pMSE/S_pMSE.
- And for the two-way utility, the synthetic dataset is evaluated with the utility tables which takes up a heatmap fashion/manner.

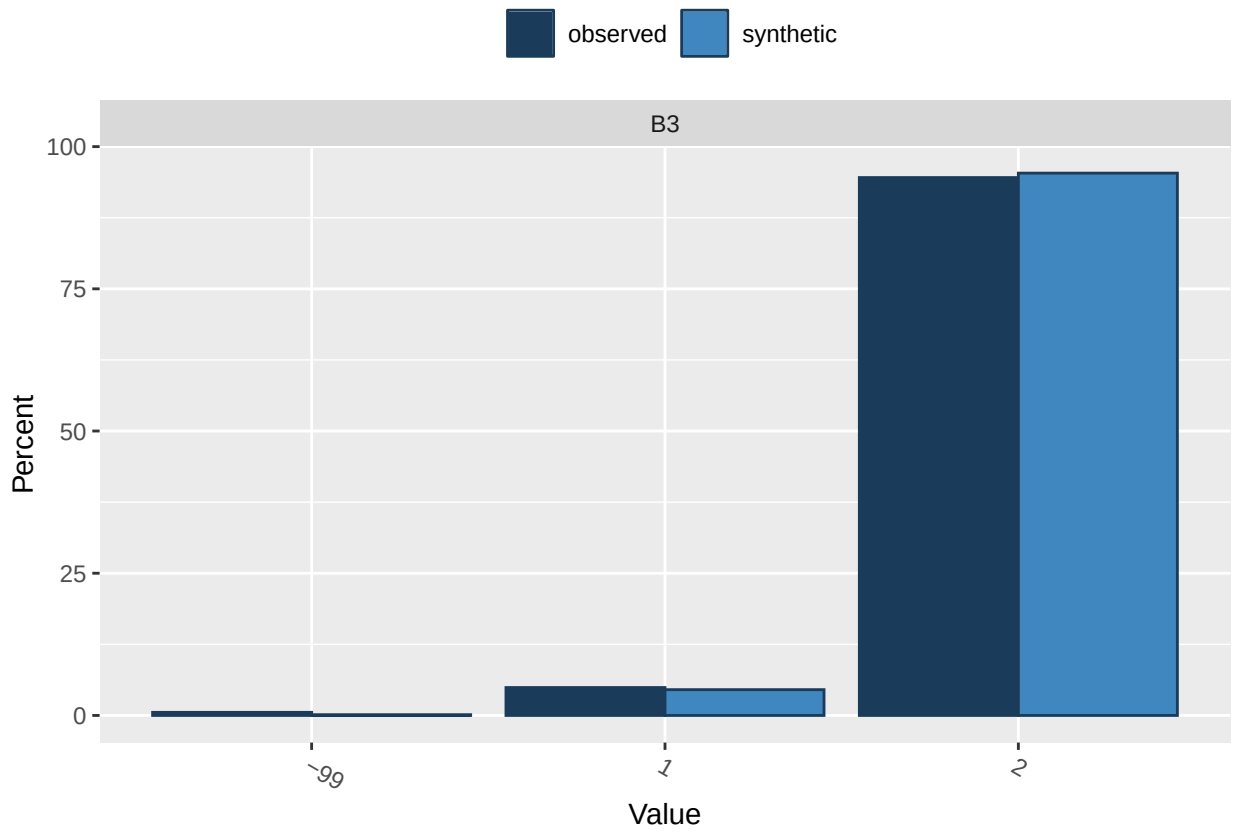
```
# # subset some example columns and try plotting the compare histograms
# # these are symptoms variables
# symptoms <- c("B3","B4","B1_1","B1_2","B1_3","B1_4","B1_5","B1_6","B1_7",
#               "B1_8","B1_9","B1_10","B1_12","B1_13",
#               "B1b_x1","B1b_x2","B1b_x3","B1b_x4","B1b_x5","B1b_x6","B1b_x7",
#               "B1b_x8","B1b_x9","B1b_x10","B1b_x12","B1b_x13","B2")
#
```

```r
# # these are testing variables
# testing <- c("B7")
#
# # Columns `B0`, `B8a`, `B15_1`, `B15_2`, `B15_3`, etc. don't exist
#
# ori_dataset_symptoms <- as.data.frame(bindori_dataset_filtered[, symptoms])
# syn_dataset_symptoms <- as.data.frame(syn_data_filtered[, symptoms])
#
# ori_dataset_testing <- as.data.frame(bindori_dataset_filtered$B7)
# syn_dataset_testing <- as.data.frame(syn_data_filtered$B7)
```

(1). one-way marginals using compare()

```r
## for var B3 -> symptoms
compare(object = data.frame(B3 = syn_data_filtered$B3),
        data = data.frame(B3 = bindori_dataset_filtered$B3),
        vars = c("B3"), cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = TRUE)
```

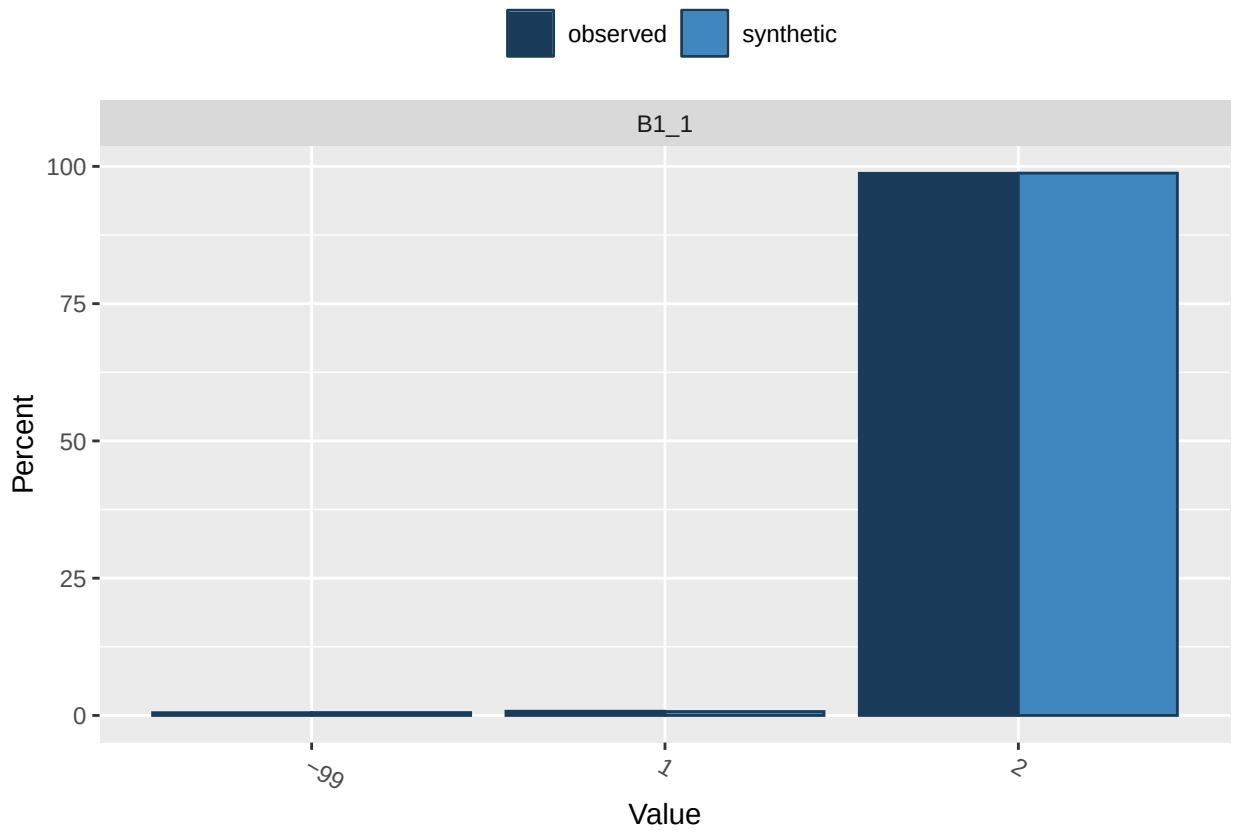
```
##
## Comparing percentages observed with synthetic
##
## $B3
##                  -99        1        2
## observed  0.5424531 4.903592 94.55396
## synthetic 0.1320000 4.531000 95.33700
```




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## B3 0.000172 214.2633
```

```
## for var B1_1 -> symptoms
compare(object = data.frame(B1_1 = syn_data_filtered$B1_1),
        data = data.frame(B1_1 = bindori_dataset_filtered$B1_1),
        vars = "B1_1", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

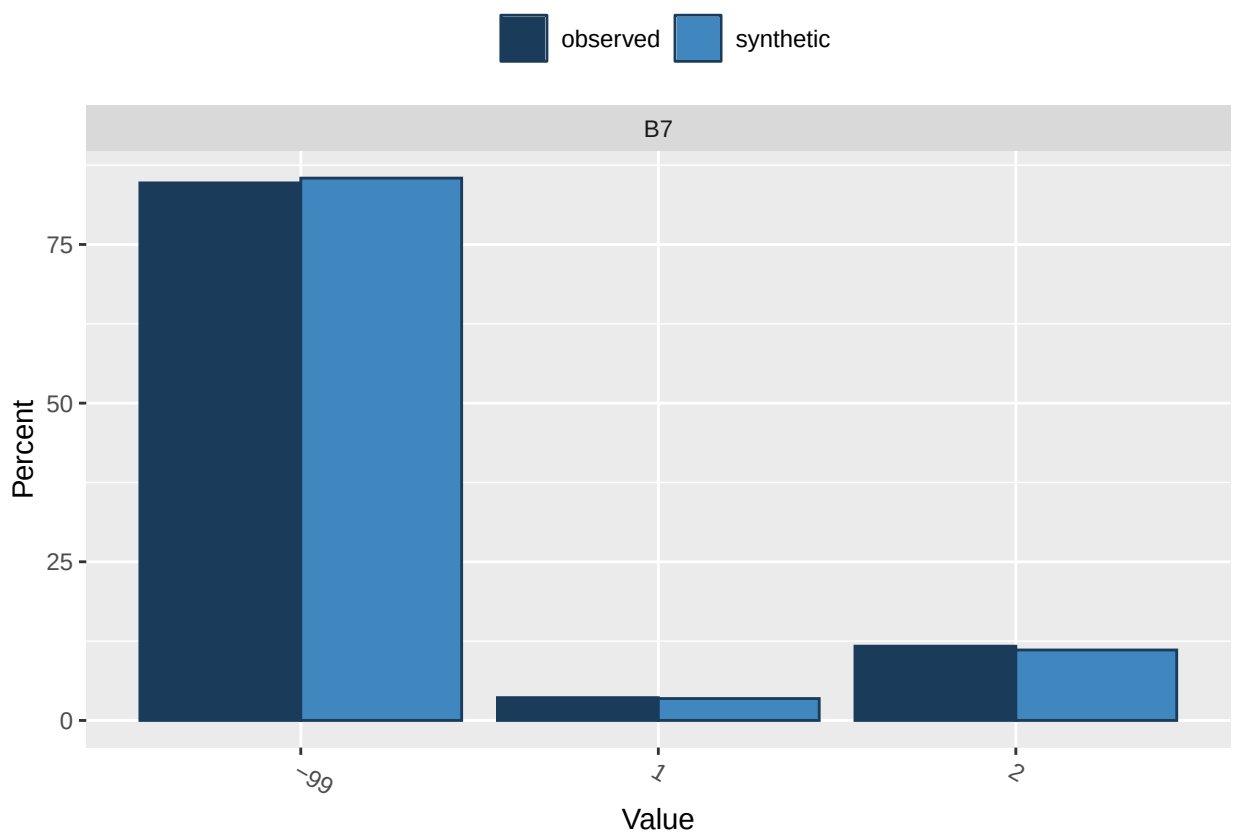
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##       pMSE   S_pMSE
## B1_1 2e-06 2.429377
```

```
## for var B7 -> testing
compare(object = data.frame(B7 = syn_data_filtered$B7),
        data = data.frame(B7 = bindori_dataset_filtered$B7),
        vars = "B7", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

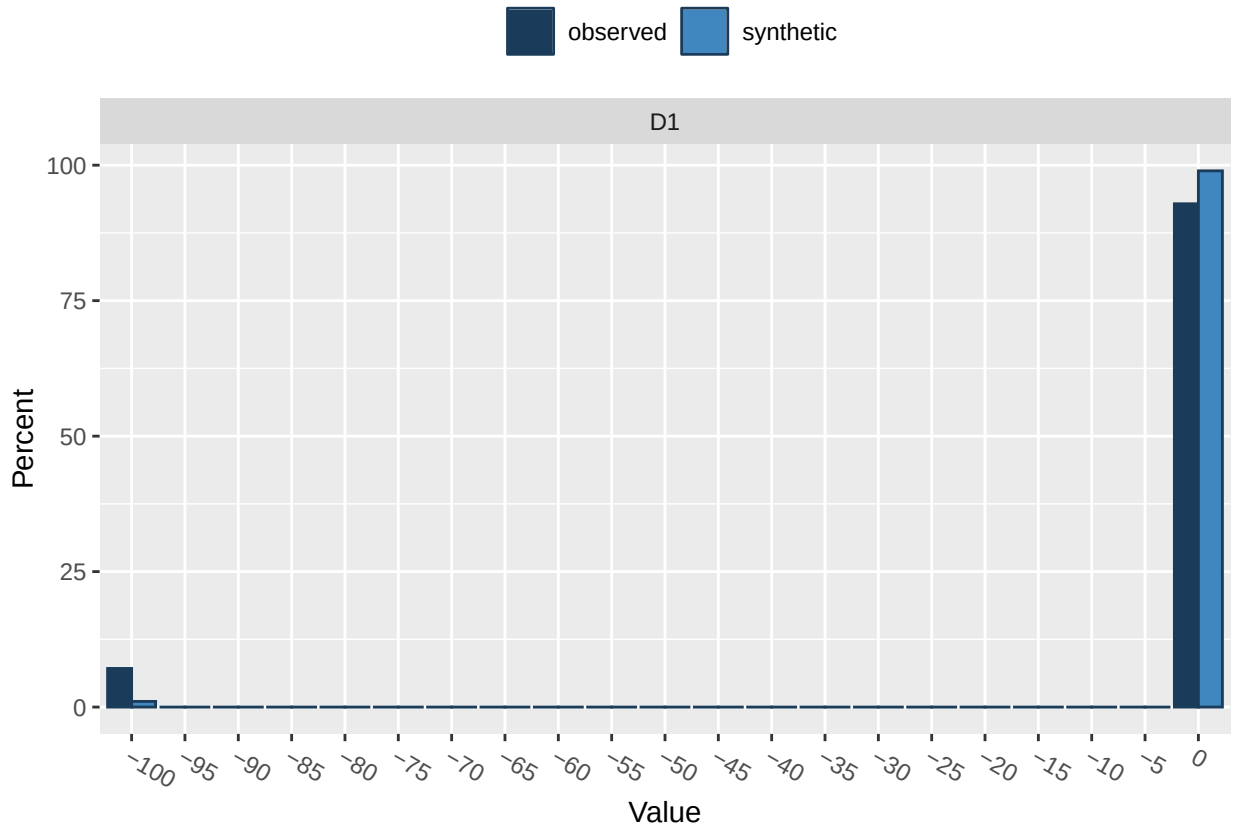
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##       pMSE   S_pMSE
## B7 1.8e-05 22.01889
```

```
## for var D1 -> module B
compare(object = data.frame(D1 = syn_data_filtered$D1),
        data = data.frame(D1 = bindori_dataset_filtered$D1),
        vars = "D1", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

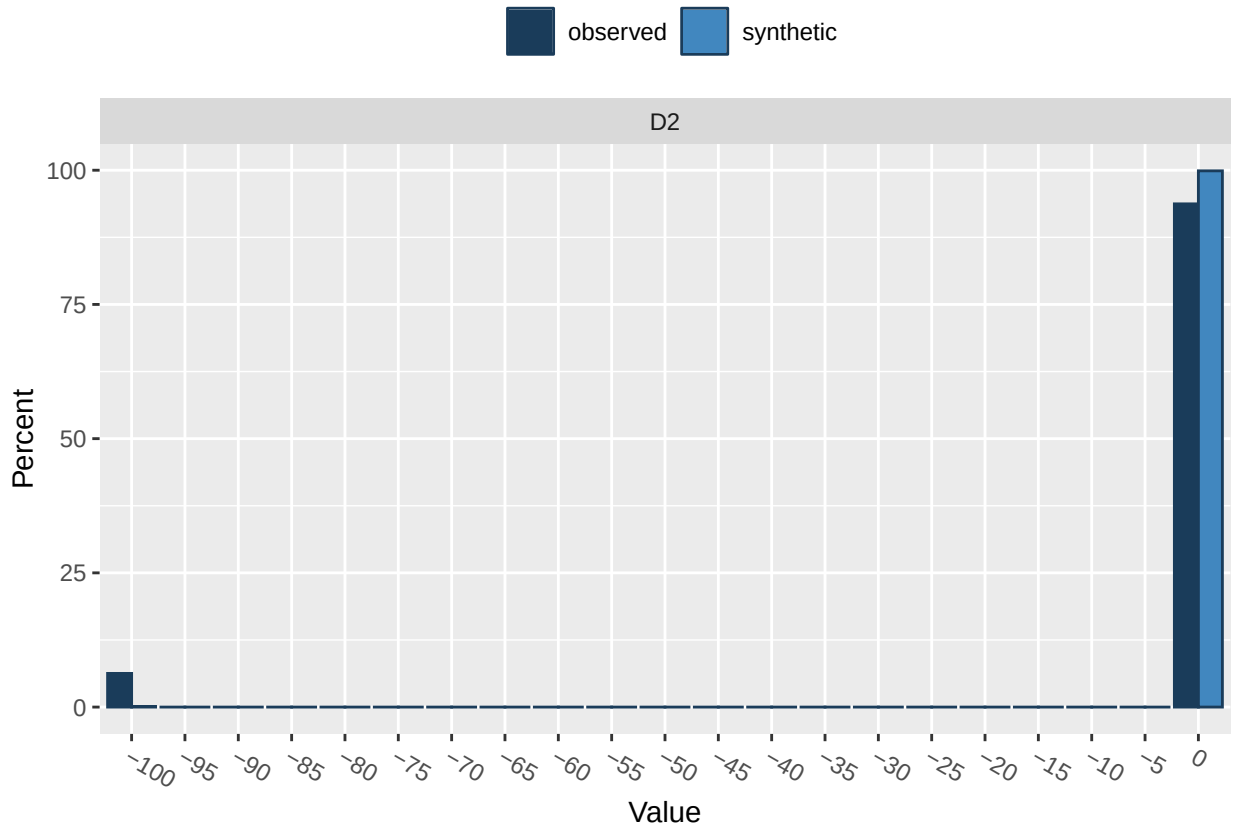
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## D1 0.000835 2076.806
```

```
## for var D2 -> module B
compare(object = data.frame(D2 = syn_data_filtered$D2),
        data = data.frame(D2 = bindori_dataset_filtered$D2),
        vars = "D2", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

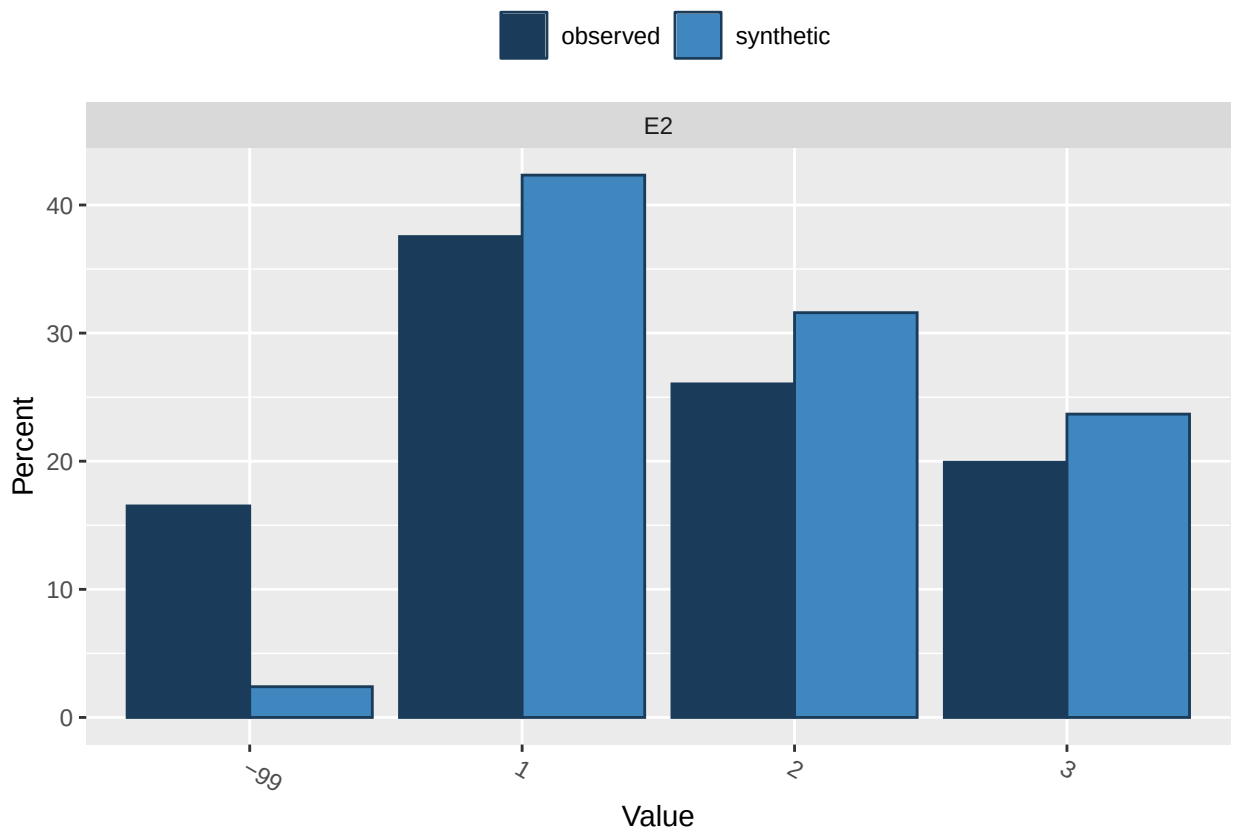
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## D2 0.000665 1654.444
```

```
## for var E2 -> demographics
compare(object = data.frame(E2 = syn_data_filtered$E2),
        data = data.frame(E2 = bindori_dataset_filtered$E2),
        vars = "E2", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

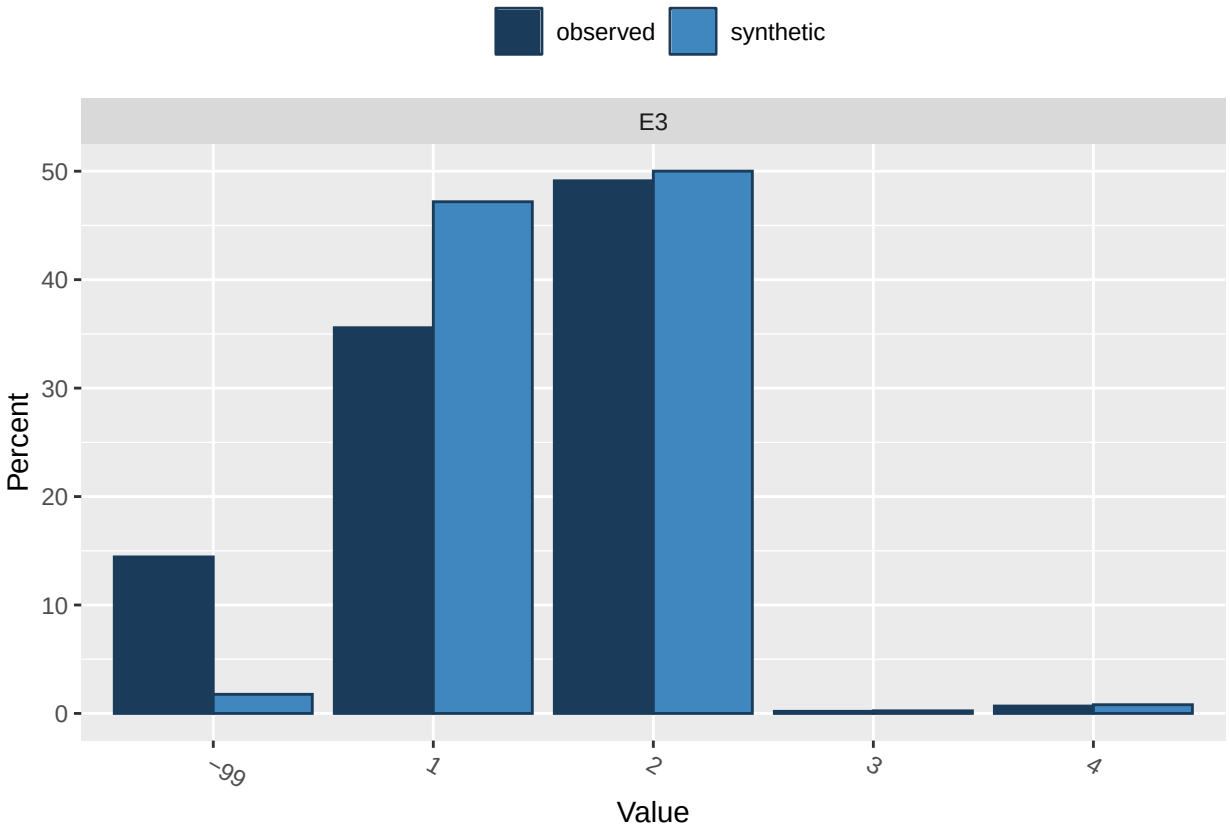
```
##
## Comparing percentages observed with synthetic
```




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## E2 0.007316 6065.024
```

```
## for var E3 -> demographics
compare(object = data.frame(E3 = syn_data_filtered$E3),
        data = data.frame(E3 = bindori_dataset_filtered$E3),
        vars = "E3", cont.na = NULL,
        msel = NULL, stat = "percents", breaks = 20,
        nrow = 2, ncol = 2, rel.size.x = 1,
        utility.stats = c("pMSE", "S_pMSE"),
        cols = c("#1A3C5A","#4187BF"),
        plot = TRUE, table = FALSE)
```

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




```
##
## Selected utility measures:
##        pMSE   S_pMSE
## E3 0.007331 4558.638
```

(2). two-way marginals with utility.tables()

In this part, we focus on the utility in the two-way manner/fashion. However, with a large number of vars to be included, we need to select a few variables to have a first glance of what does the `utility.tables` look like.

```
## filter out a few variables to run the evaluation
## pls make sure that the vars selected are in alignment
## with the one-way maginal ones
selected_cols <- c("B3", "B1_1", "B7", "D1", "D2", "E2", "E3")
syn_select_vars <- syn_data_filtered[, selected_cols]
bindori_select_vars <- bindori_dataset_filtered[, selected_cols]
```

```
## S3 method for class 'data.frame'
utility.twoway <- utility.tables(object = data.frame(syn_select_vars),
                                 data = data.frame(bindori_select_vars),
                                 tables = "twoway",
                                 tab.stats = c("pMSE", "S_pMSE"),
                                 plot.stat = "S_pMSE", plot = TRUE,
                                 print.tabs = TRUE)
```

Now, we print the results out with the heatmap-like output plot.

```
utility.twoway$utility.plot
```