Synthetic Data Description

Robert Arbon

08/04/2019

Introduction

In order to synthesize the data we are using an R package called *SynthPop*. The vignette is extremely informative and gives many good references. The documentation can be found here.

The method is very similar to Multiple Imputation by Chained Equations (MICE) except that missing values are treated as distinct and valid value (rather than something to be imputed). Please refer to the vignette section 2 for a description of the method.

Our synthesis method

The method we have employed is as follows:

1. We use a Classification and Regression Tree (i.e. using `syn(method=cart)`) to build the conditional distributions for each variable, with the following parameters:
2. we used proper synthesis, i.e. `proper=TRUE`
3. smoothing was applied to the continuous variables: `iq`, `height_16` and `weight_16`.
4. only one replicant was produced, `m=1`.
5. The random seed was `seed = 8308313`.
6. The visit sequence was just the order of the dataframe.
7. All ther parameters were set to their defaults.
8. Once this was done we removed any observations that were in the original dataset using `anti_join` from `dplyr`.

For transparency we'll go through this method here:

Synthesize data:

The following script was run to synthesize the data on a local compute cluster. A random seed was supplied as an argument:

```
library(synthpop)
library(tidyverse)

# Take a random seed from cluster job ID.  Only the first part is an integer.
args <- commandArgs(trailingOnly=TRUE)
my_seed <- strsplit(args[1], "\\.")[[1]][1]
my_seed <- as.integer(my_seed)
print(my_seed)

# Load data
df <- read_rds('data/maps-original.rds')

# Synthesize
df_syn <- syn(df, method="cart", m = 1,proper=T, smoothing=list("iq"="density", "height_16"="density",
# Write out
write_rds(x=df_syn, path=paste('output/cart/maps-synds-', my_seed, '.rds', sep=""))
```

Load data

At this point the above script has been run and the output downloaded.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(synthpop)
```

```
## Loading required package: lattice

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: nnet
```

```
sds <- read_rds('output/cart_synth_ds/maps-synds-8308694.rds')
ods <- read_rds('data/maps-original.rds')
```

We can check the random seed and the call signature to make sure we've got the correct data.

```
sds$seed
```

```
## [1] 8308694
```

```
sds$call
```

```
## syn(data = df, method = "cart", m = 1, proper = T, smoothing = list(iq = "density",
##     height_16 = "density", weight_16 = "density"), seed = my_seed)
```

Drop duplicates

The `sds` object is a SynthPop object which contains the synthetic dataframe amongst all the other associated (meta)data. Here we use an `anti_join` to select those rows of the synthetic dataframe that are an exact match to those in the original dataframe.

As this is perhaps not very intuitive, let's have an simple example:

```
table1<-data.frame(id=seq(1,6), vals=rnorm(6))
table2<-table1[c(1,3,4),]
table1
```

```
##   id        vals
## 1  1  0.01991039
## 2  2  1.78028122
## 3  3 -0.45165838
## 4  4  0.24391266
## 5  5  0.67606484
## 6  6  0.46407175
```

```r
table2
```

```
##   id        vals
## 1  1  0.01991039
## 3  3 -0.45165838
## 4  4  0.24391266
```

```r
anti_join(table1, table2)
```

```
## Joining, by = c("id", "vals")
```

```
##   id      vals
## 1  2 1.7802812
## 2  5 0.6760648
## 3  6 0.4640718
```

So we can use this to overwrite the synthetic dataset with one which has no rows in common with the original dataset:

```r
# check the original dimensions
dim(sds$syn)
```

```
## [1] 14619    82
```

```r
# Overwrite
sds$syn <- anti_join(sds$syn, ods, by=colnames(ods))
# Check dimensions of new dataset
dims <- dim(sds$syn)
# k is the number of observations
sds$k <- dims[1]
dims
```

```
## [1] 13734    82
```

Comparisons

In this section we'll compare the dataset by running checking:

1. The number of complete cases on a selection of the variables (the outcomes and exposures)
2. Some logistic/multinomial regressions on some of the binary/ternary variables.
3. The marginal distributions of each variable (these are presented as charts and placed at the end)

Complete cases

```r
# The important variables
important_vars <- c('has_dep_diag', 'prim_diag', 'secd_diag', 'dep_thoughts',
                    'dep_score', 'comp_week', 'comp_wend')
```

```r
# This calculates the number of complete cases
n_comp_cases <- function(df){
  dims <- df %>%
      dplyr::select(important_vars) %>%
      dplyr::filter(complete.cases(.)) %>%
      dim()
  dims[1]
}

print(sprintf("Real DS complete cases: %4d", n_comp_cases(ods)))
```

```
## [1] "Real DS complete cases: 1206"
```

```r
print(sprintf("Synthetic DS complete cases: %d",n_comp_cases(sds$syn)))
```

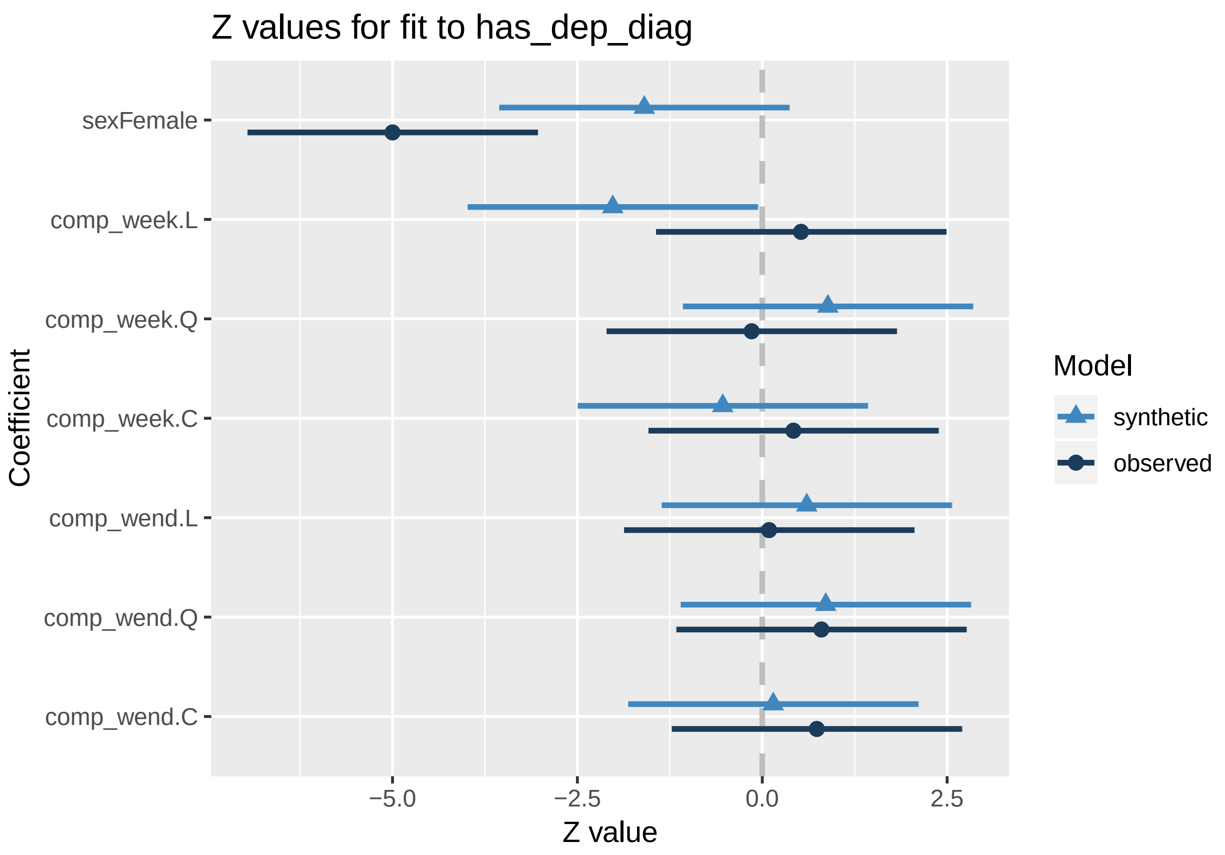
```
## [1] "Synthetic DS complete cases: 1268"
```

Regressions

We'll use the SynthPop functions to run and compare the regressions. First we'll regress the ICD diagnosis indicator against `sex`, `comp_week` and `comp_wend`.

```r
# We'll need these variables for the other regressions:
set.seed(42)
df <- sds$syn
col_names <- colnames(df)
n_cols <- length(col_names)
```

```r
# Fit the regression
mod <- glm.synds(formula=has_dep_diag ~ sex + comp_week + comp_wend,
                 family="binomial", data=sds)

# Get the comparison:
comp<- synthpop::compare(mod, ods, print.coef=F)
print(comp$ci.plot)
```



Now we'll do some random regressions: 1. We'll randomly select columns, the first column will be the dependent variable. 2. Check whether the dependent variable has 2 or 3 unique levels (excluding NAs) 3. Check whether the covariates have 2 or more levels (excluding NAs). 4. Fit the model. 5. Plot the Z scores and CIs for each covariate on the same plot. 6. Repeate 10 times.

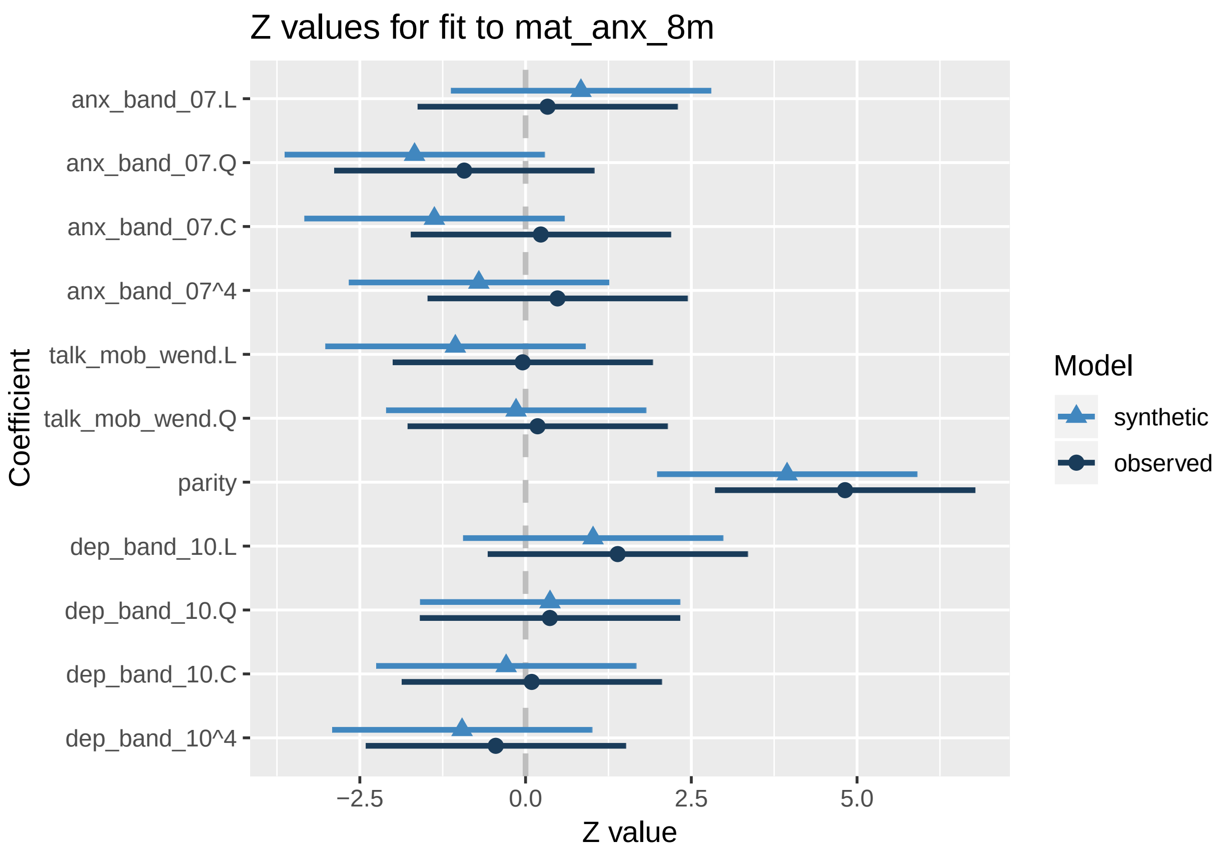
Note: Some comparisons won't work because different number of levels in the complete cases in original and synthetic data set. We'll ignore these.

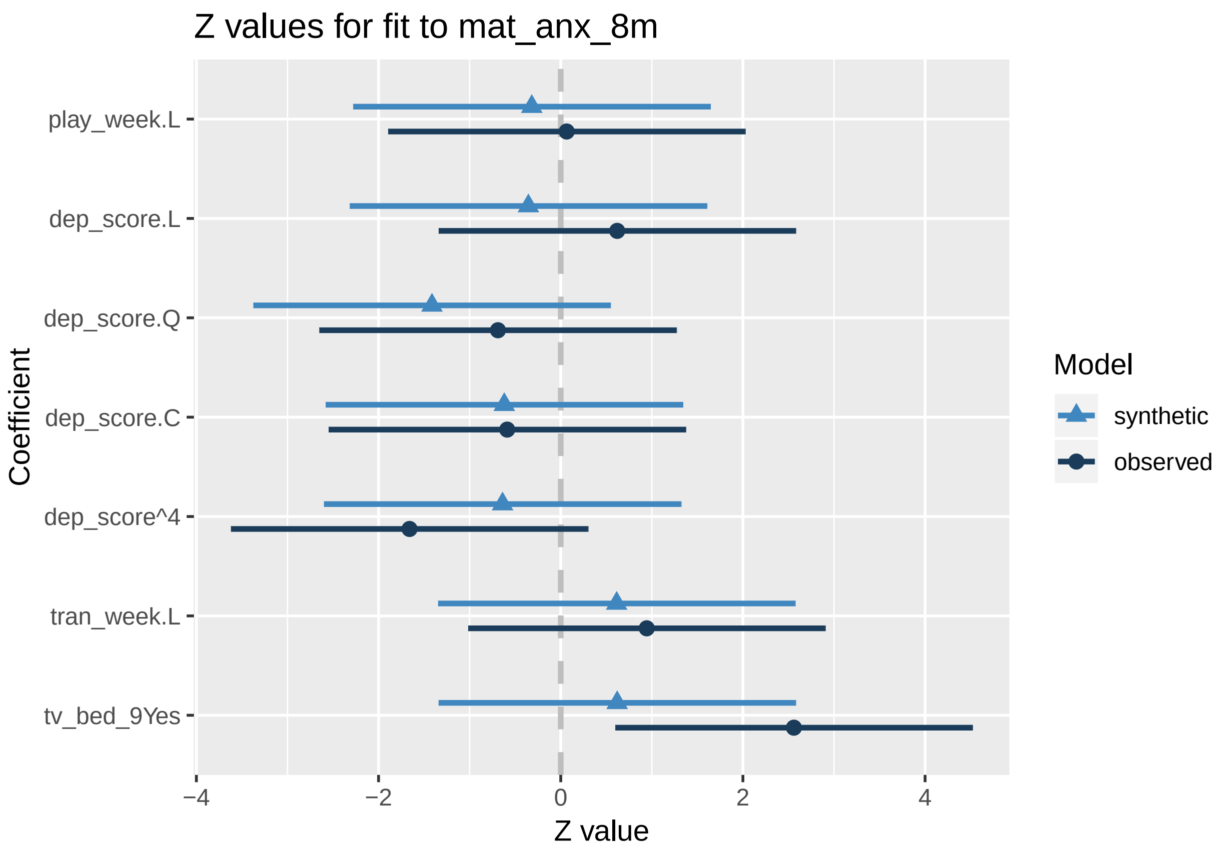
First, here are some helper functions:

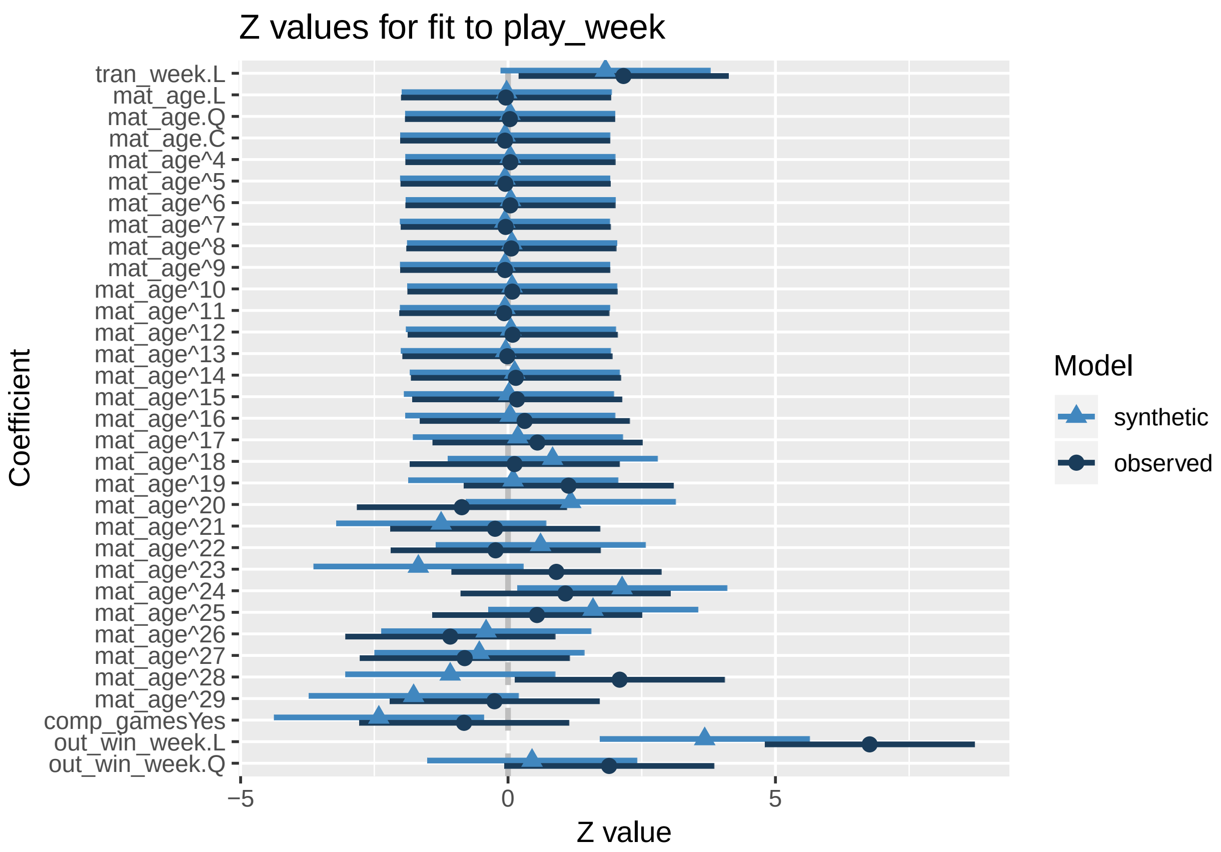
```
# This tests whether the number of cases is greater than 2 for all variables.
# And whether the first variable has less than 4 cases (allowing for ternary outcomes)
is_valid_selection <- function(df, col_nums){
  n_vals <- sapply(col_nums, function(x) length(unique(df[,x])))
  flag <- (min(n_vals) > 2) & (n_vals[1] < 4)
  flag
}

# Creates a formula for a single col name for the DV, and a vector of col names for the IVs.
create_formula <- function(dv, ivs){
      form <- paste(sapply(ivs[2:length(ivs)], function(x) sprintf(' + %s', x)), collapse="")
      form <- paste(dv, sprintf('~ %s', ivs[1]), form)
      form <- as.formula(form)
      form
}
```

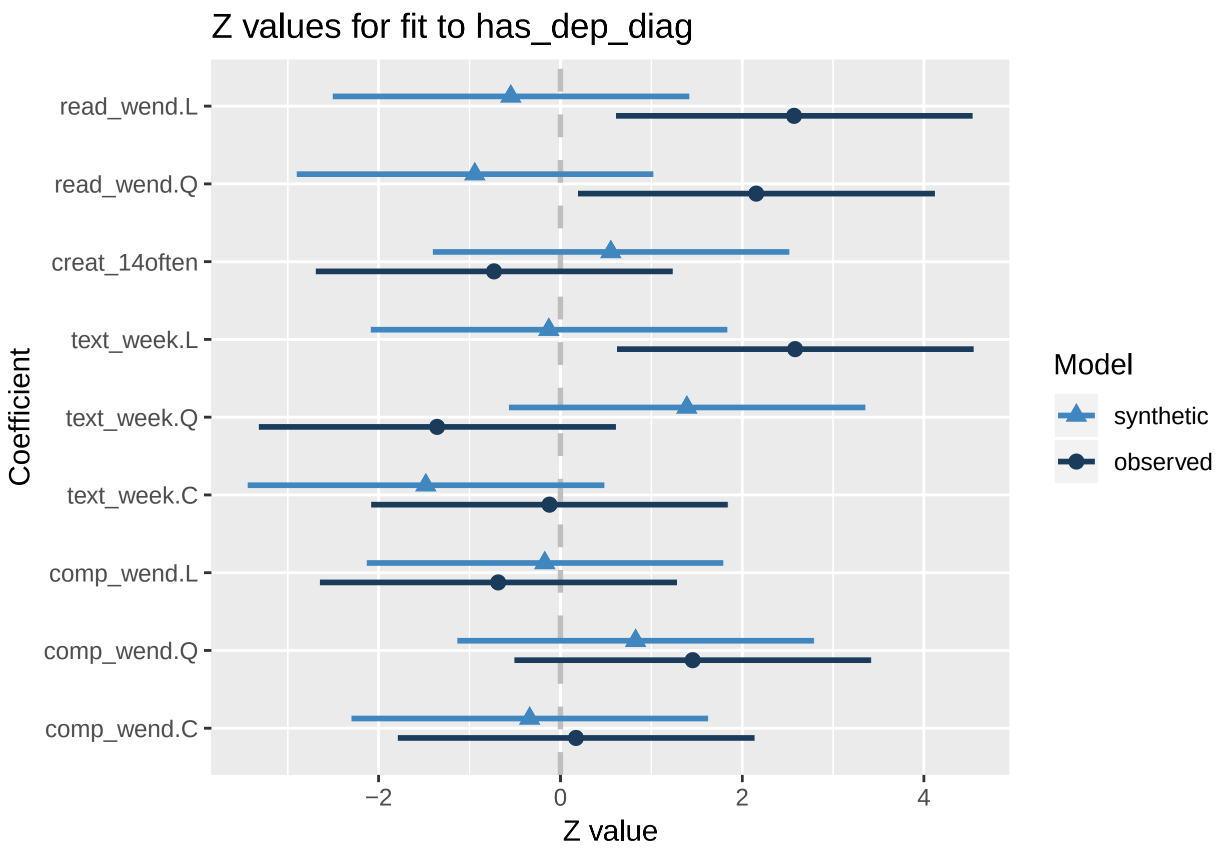
Now let's loop!

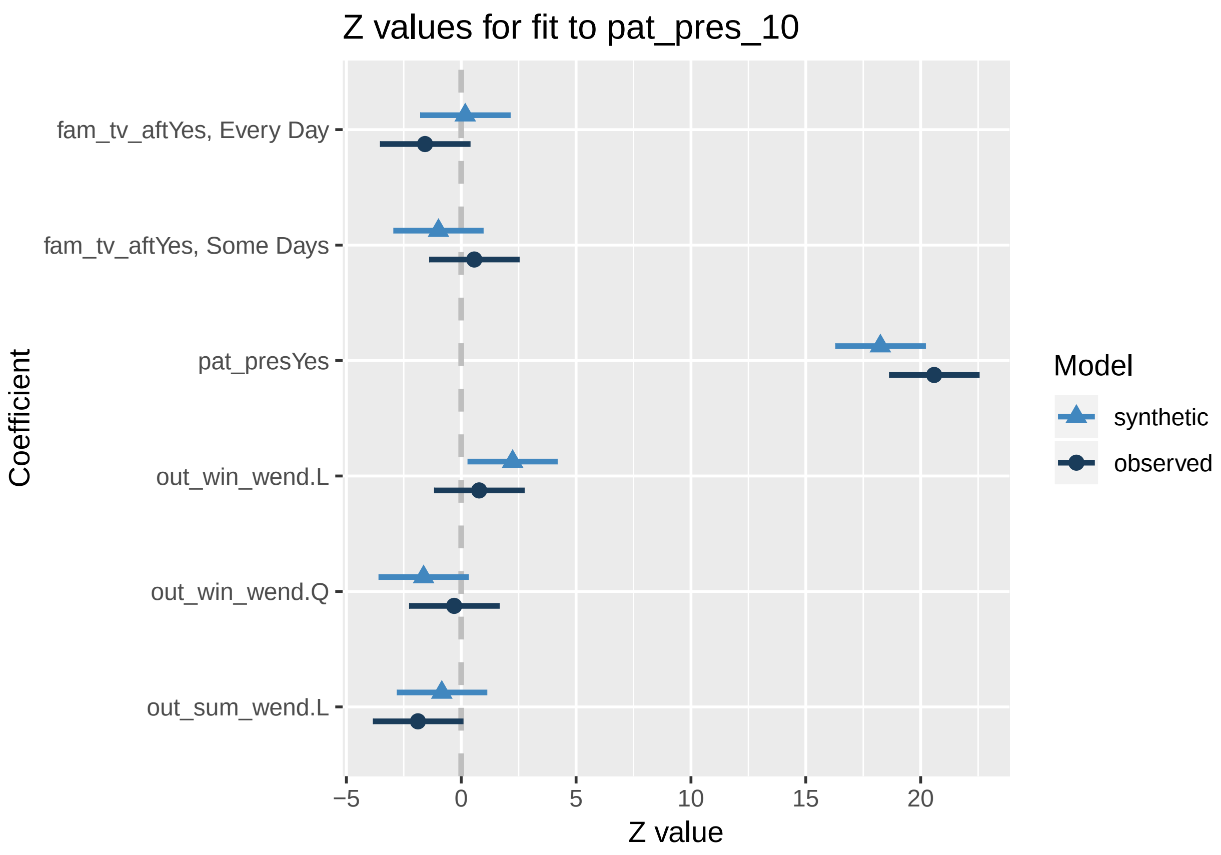
```r
count <- 1

while(count <= 10){

  # Select some random columns
  cols <- sample(n_cols, 5)
  # Check variables have more than 3 unique values and the first column has less than 4
  flag <- is_valid_selection(df, cols)

  if (flag){
    tryCatch({

      # Get the column names
      c_names <- sapply(cols, function(x) col_names[x])

      # Generate the formula for the regression
      form <- create_formula(c_names[1], c_names[2:length(c_names)])

      # Fit the regression
      mod <- glm.synds(formula=form, family="binomial", data=sds)

      # Get the comparison:
      comp<- synthpop::compare(mod, ods, print.coef=F)
      print(comp$ci.plot)
      count <- count + 1

    }, warning=function(e){
      # Some comparisons won't work because different number of complete cases in
      # original and synthetic data set.Just silently fail and move on.
    })
  }
}
```

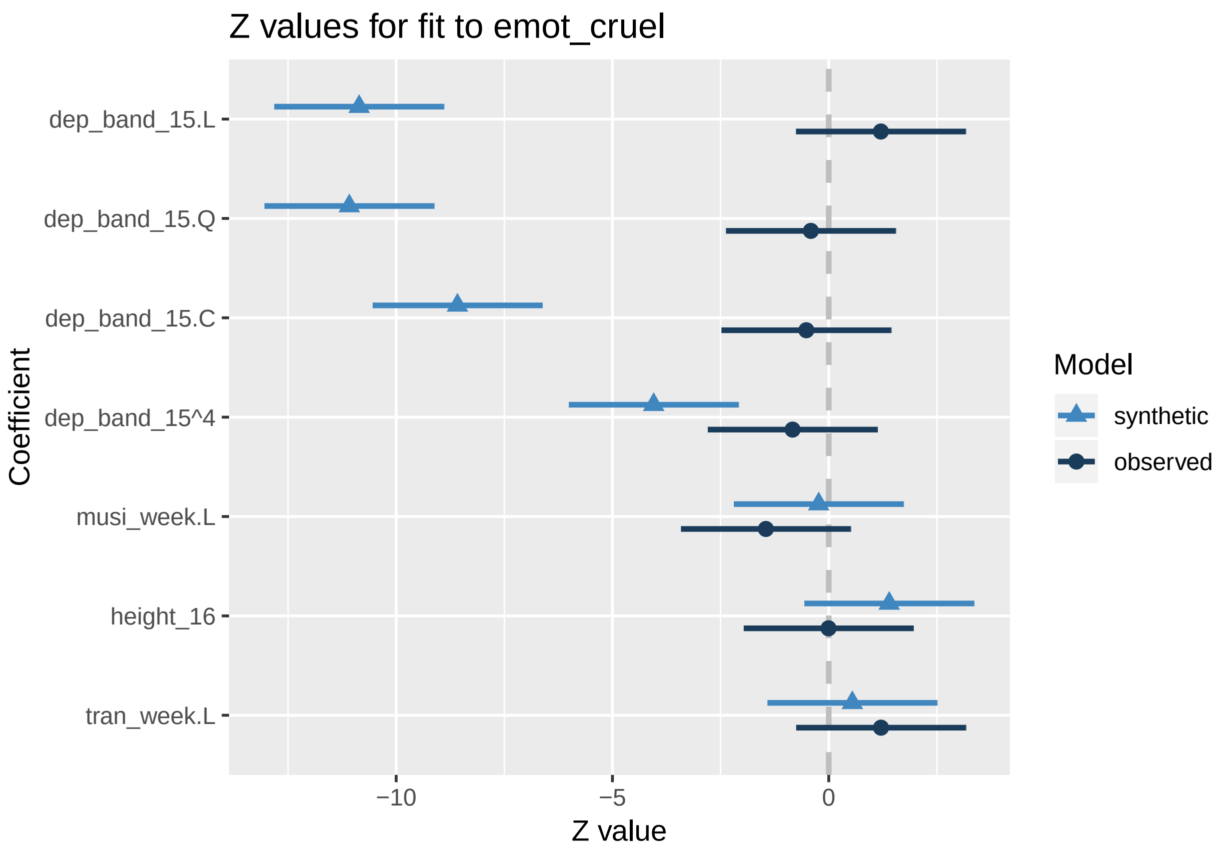

Z values for fit to mat_anx_8m
Coefficient
anx_band_07.L
anx_band_07.Q
anx_band_07.C
anx_band_07^4
talk_mob_wend.L
talk_mob_wend.Q
parity
dep_band_10.L
dep_band_10.Q
dep_band_10.C
dep_band_10^4
−2.5
0.0
2.5
5.0
Z value
Model
synthetic
observed

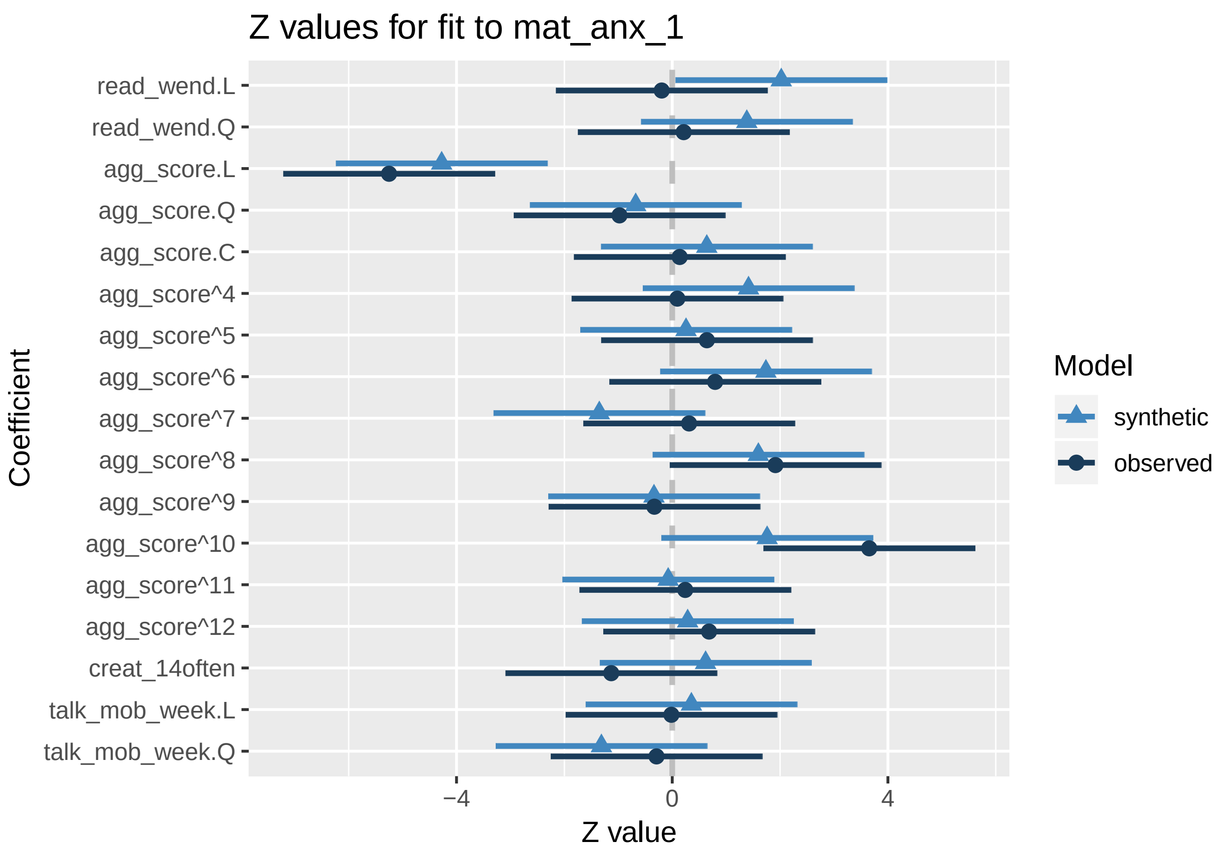

Z values for fit to mat_anx_8m
Coefficient
play_week.L
dep_score.L
dep_score.Q
dep_score.C
dep_score^4
tran_week.L
tv_bed_9Yes
−4
−2
0
2
4
Z value
Model
synthetic
observed

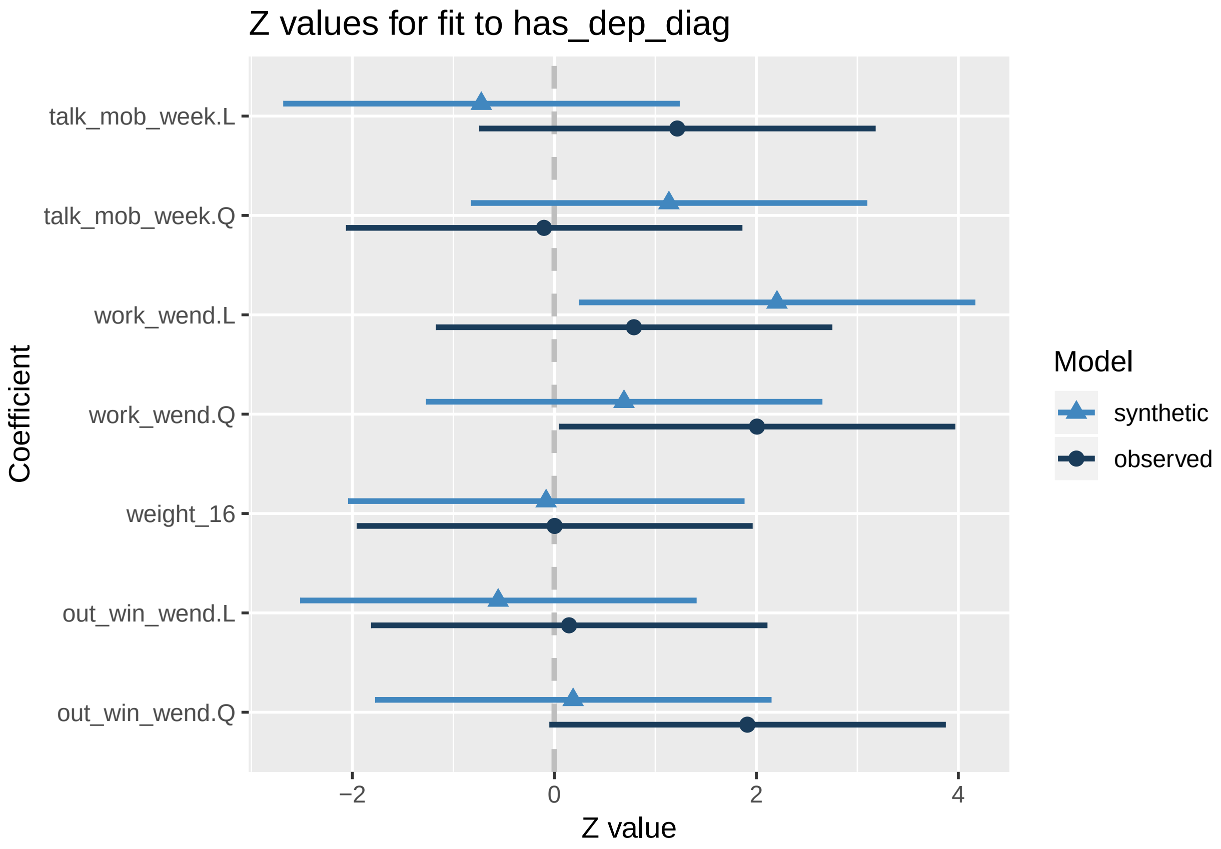

Z values for fit to play_week
Coefficient
tran_week.L
mat_age.L
mat_age.Q
mat_age.C
mat_age^4
mat_age^5
mat_age^6
mat_age^7
mat_age^8
mat_age^9
mat_age^10
mat_age^11
mat_age^12
mat_age^13
mat_age^14
mat_age^15
mat_age^16
mat_age^17
mat_age^18
mat_age^19
mat_age^20
mat_age^21
mat_age^22
mat_age^23
mat_age^24
mat_age^25
mat_age^26
mat_age^27
mat_age^28
mat_age^29
comp_gamesYes
out_win_week.L
out_win_week.Q
−5
0
5
Z value
Model
synthetic
observed

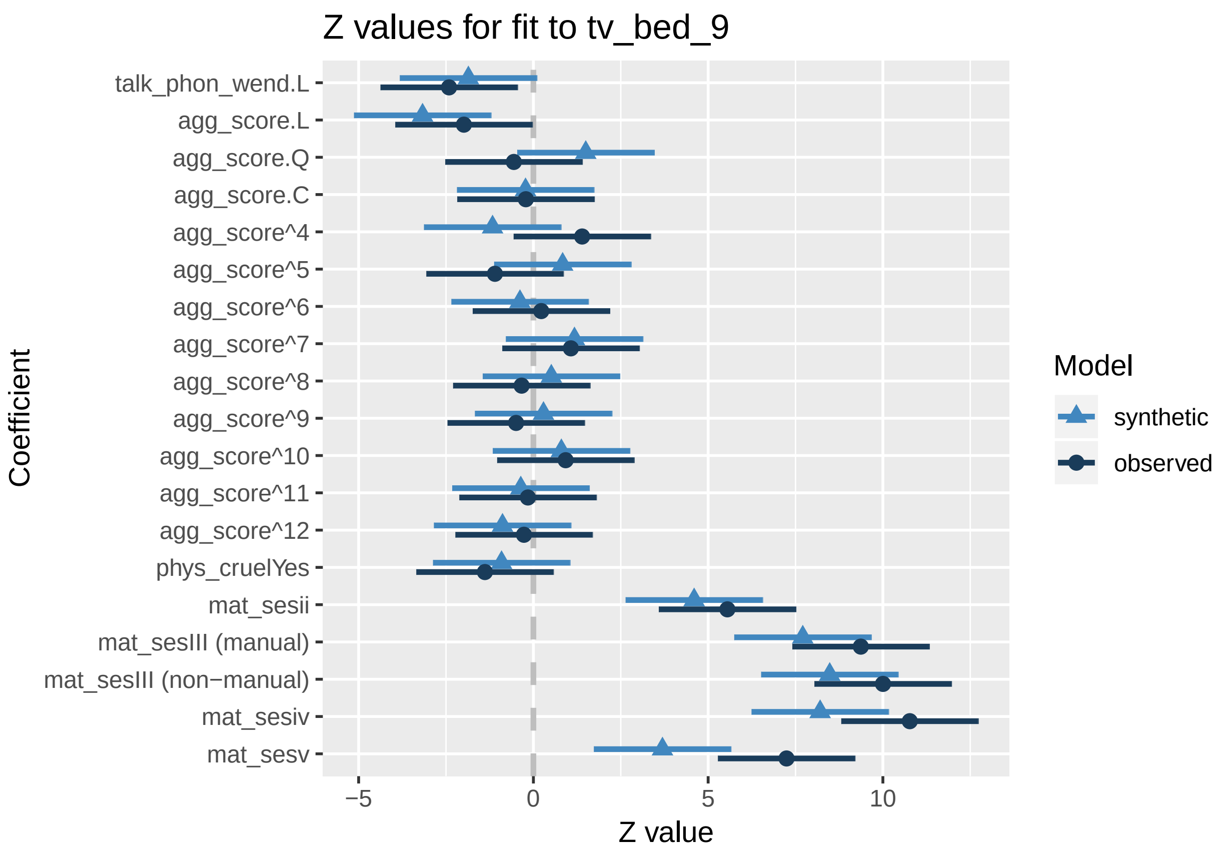

Z values for fit to has_dep_diag
Coefficient
read_wend.L
read_wend.Q
creat_14often
text_week.L
text_week.Q
text_week.C
comp_wend.L
comp_wend.Q
comp_wend.C
−2
0
2
4
Z value
Model
synthetic
observed


Z values for fit to pat_pres_10
Coefficient
fam_tv_aftYes, Every Day
fam_tv_aftYes, Some Days
pat_presYes
out_win_wend.L
out_win_wend.Q
out_sum_wend.L
−5
0
5
10
15
20
Z value
Model
synthetic
observed

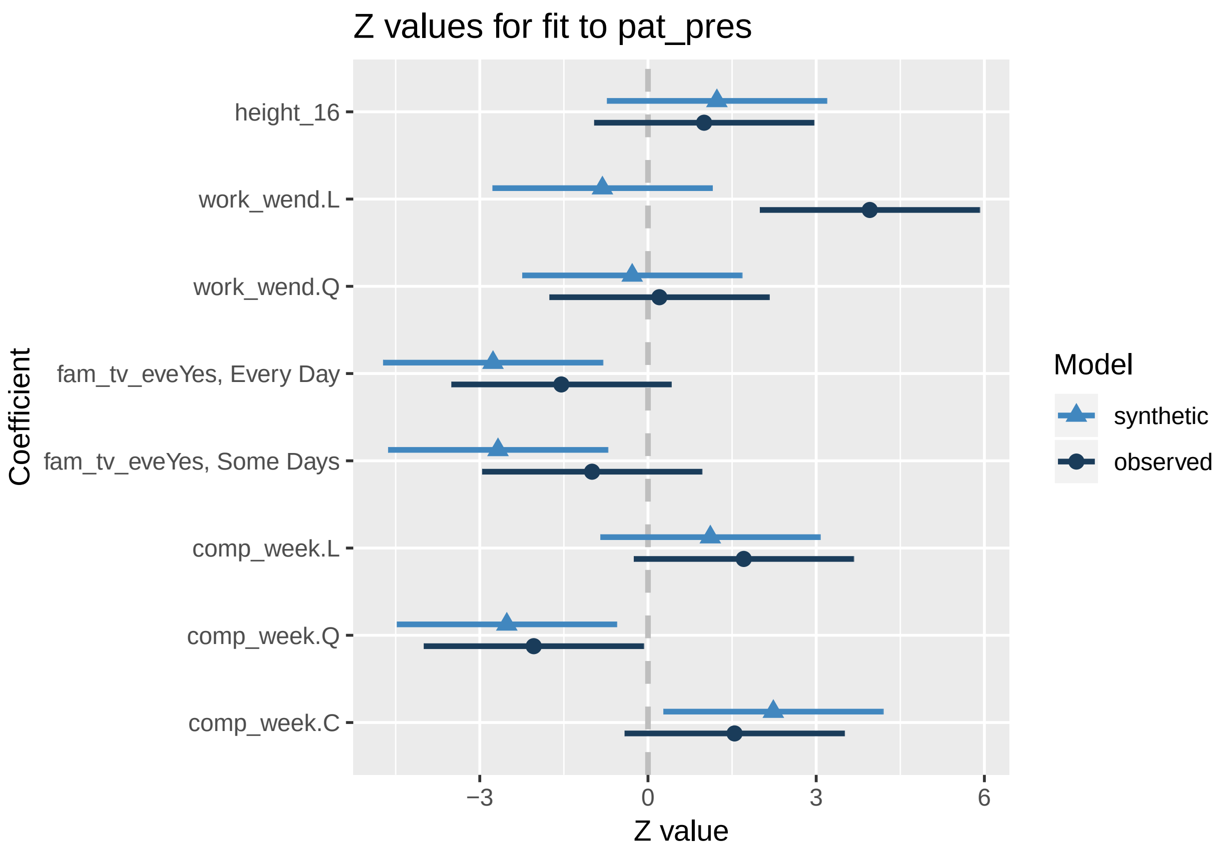

Z values for fit to emot_cruel
Coefficient
dep_band_15.L
dep_band_15.Q
dep_band_15.C
dep_band_15^4
musi_week.L
height_16
tran_week.L
−10
−5
0
Z value
Model
synthetic
observed


Z values for fit to mat_anx_1
read_wend.L
read_wend.Q
agg_score.L
agg_score.Q
agg_score.C
agg_score^4
agg_score^5
agg_score^6
agg_score^7
agg_score^8
agg_score^9
agg_score^10
agg_score^11
agg_score^12
creat_14often
talk_mob_week.L
talk_mob_week.Q
Coefficient
−4
0
4
Z value
Model
synthetic
observed


Z values for fit to has_dep_diag
talk_mob_week.L
talk_mob_week.Q
work_wend.L
work_wend.Q
weight_16
out_win_wend.L
out_win_wend.Q
Coefficient
−2
0
2
4
Z value
Model
synthetic
observed



Z values for fit to tv_bed_9
talk_phon_wend.L
agg_score.L
agg_score.Q
agg_score.C
agg_score^4
agg_score^5
agg_score^6
agg_score^7
agg_score^8
agg_score^9
agg_score^10
agg_score^11
agg_score^12
phys_cruelYes
mat_sesii
mat_sesIII (manual)
mat_sesIII (non−manual)
mat_sesiv
mat_sesv
Coefficient
−5
0
5
10
Z value
Model
synthetic
observed

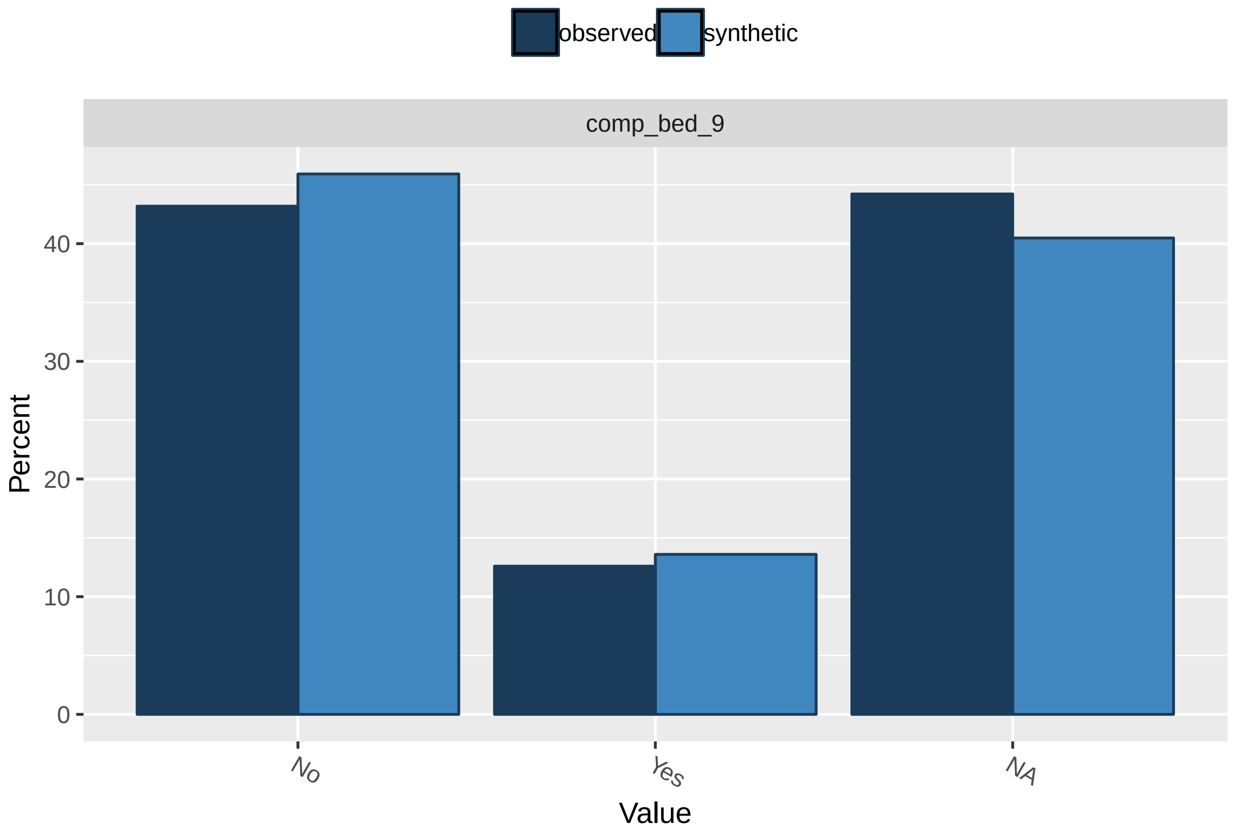


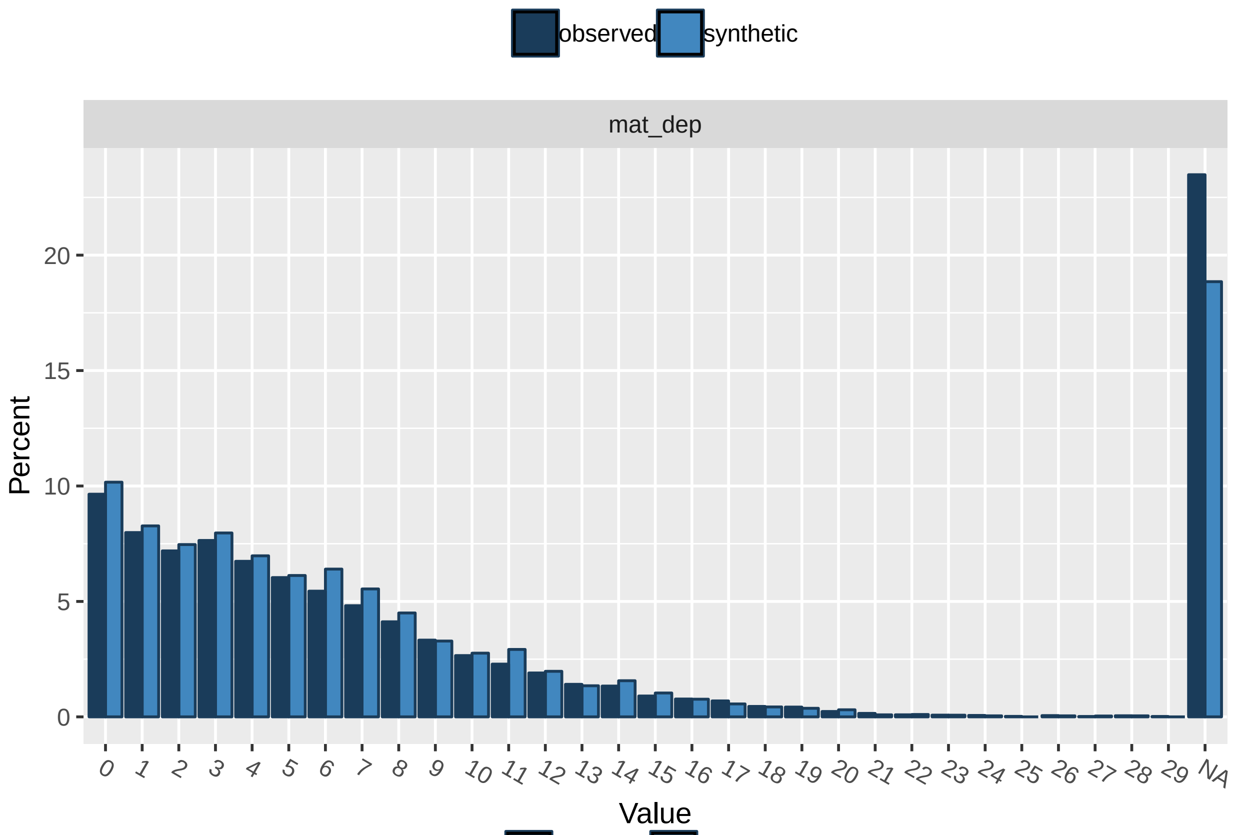
Z values for fit to pat_pres
height_16
work_wend.L
work_wend.Q
fam_tv_eveYes, Every Day
fam_tv_eveYes, Some Days
comp_week.L
comp_week.Q
comp_week.C
Coefficient
−3
0
3
6
Z value
Model
synthetic
observed

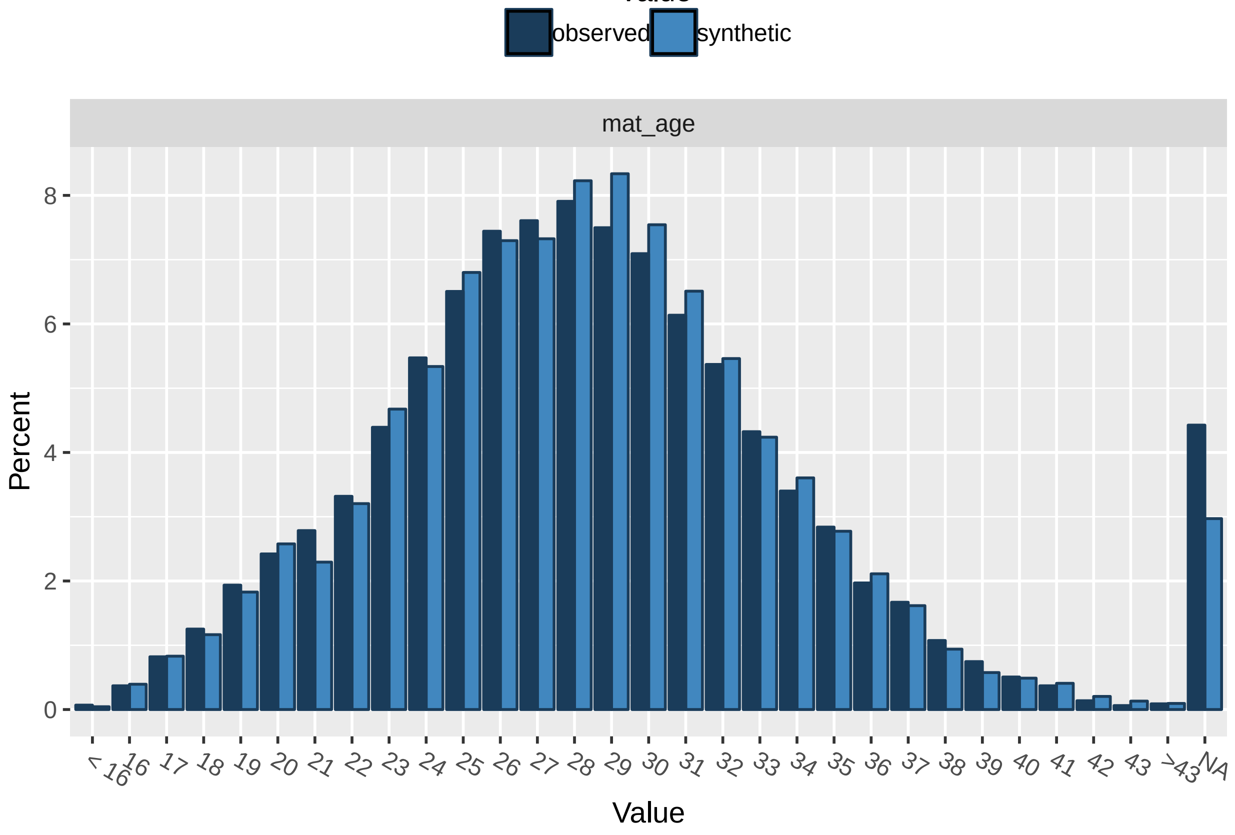
Marginal distributions

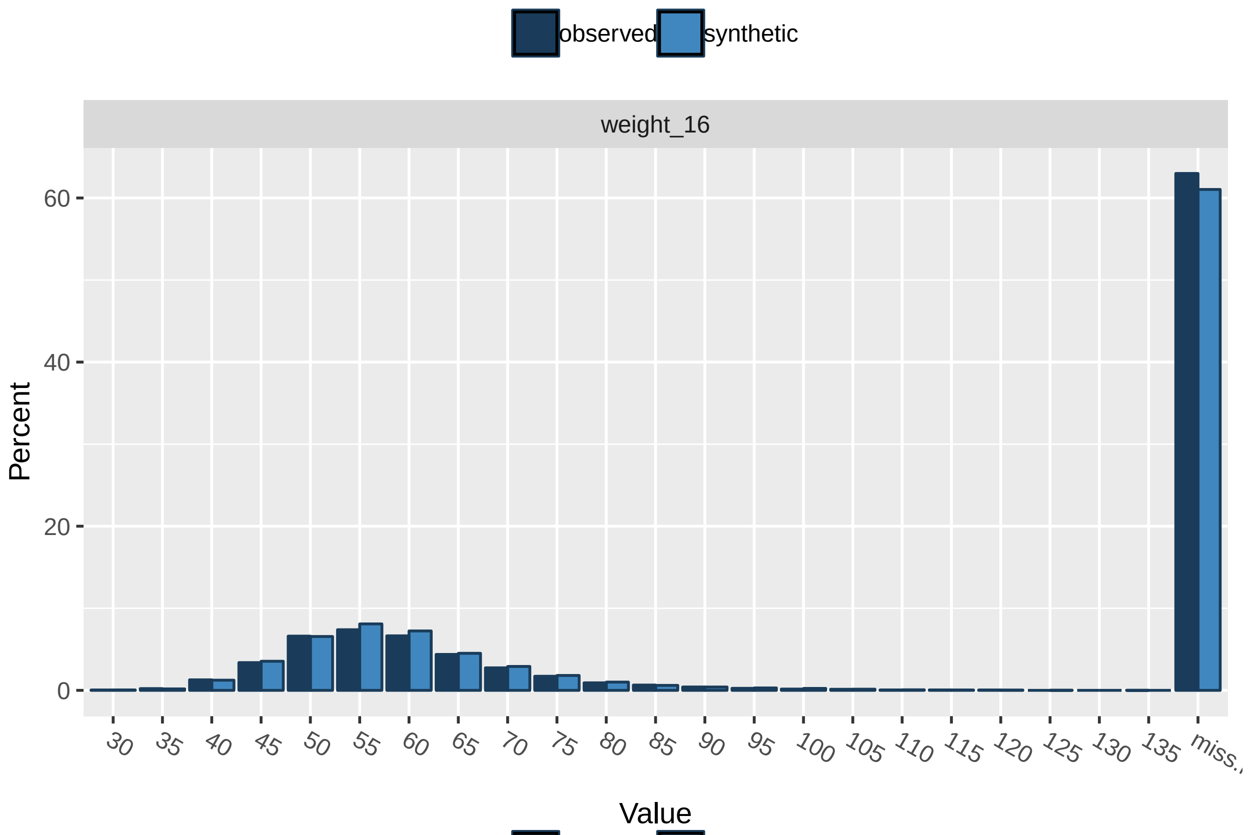
Here are the comparisons of marginal distributions.

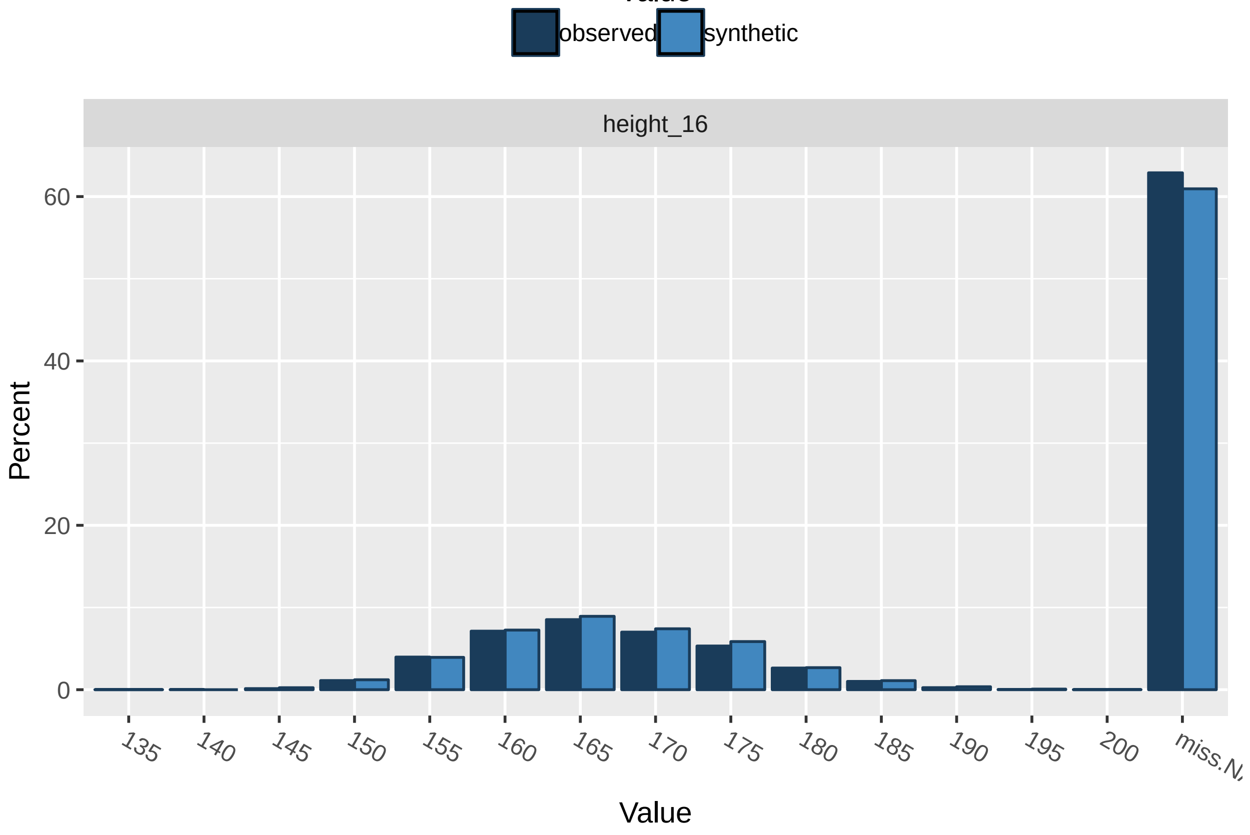
```r
for (colname in colnames(ods)){
  comparison <- synthpop::compare(sds,ods,colname)
  plt <- comparison$plots
  tbl <- comparison$tables
  plot(plt)
}
```

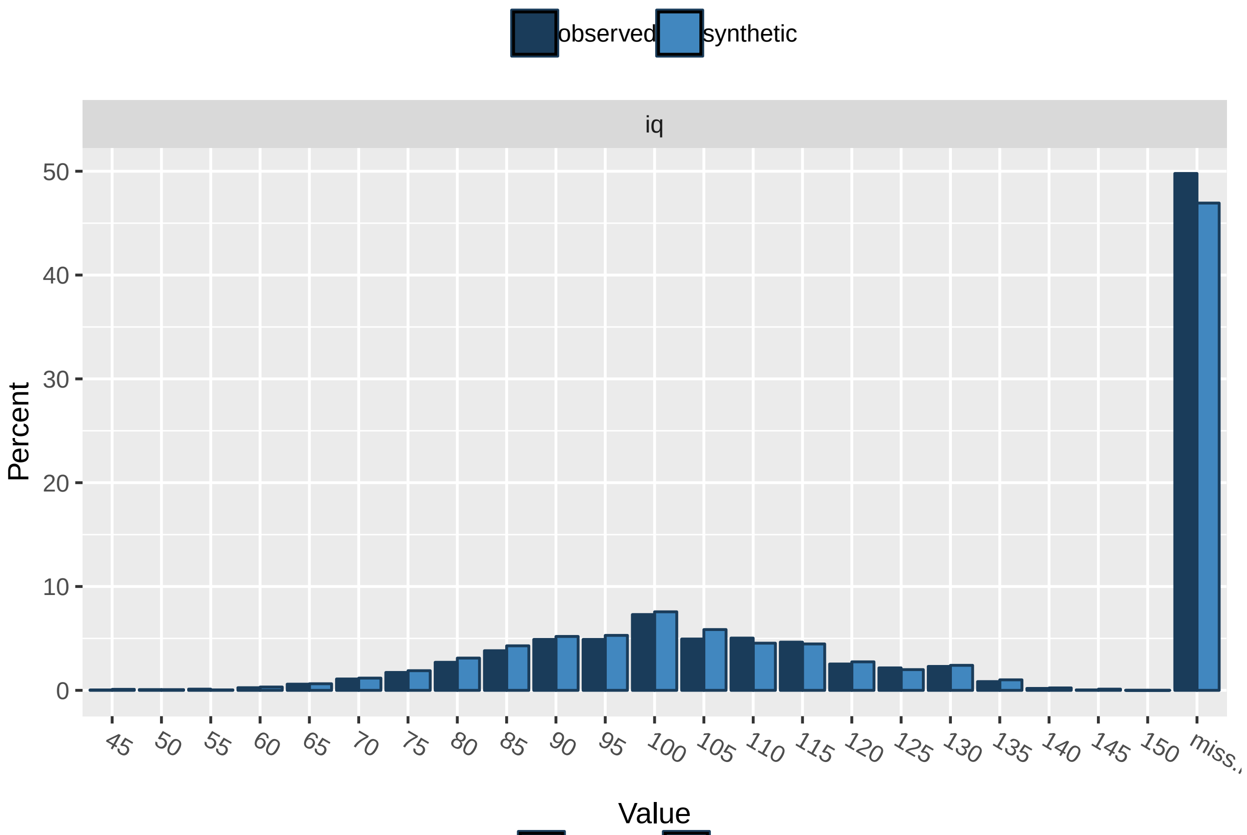


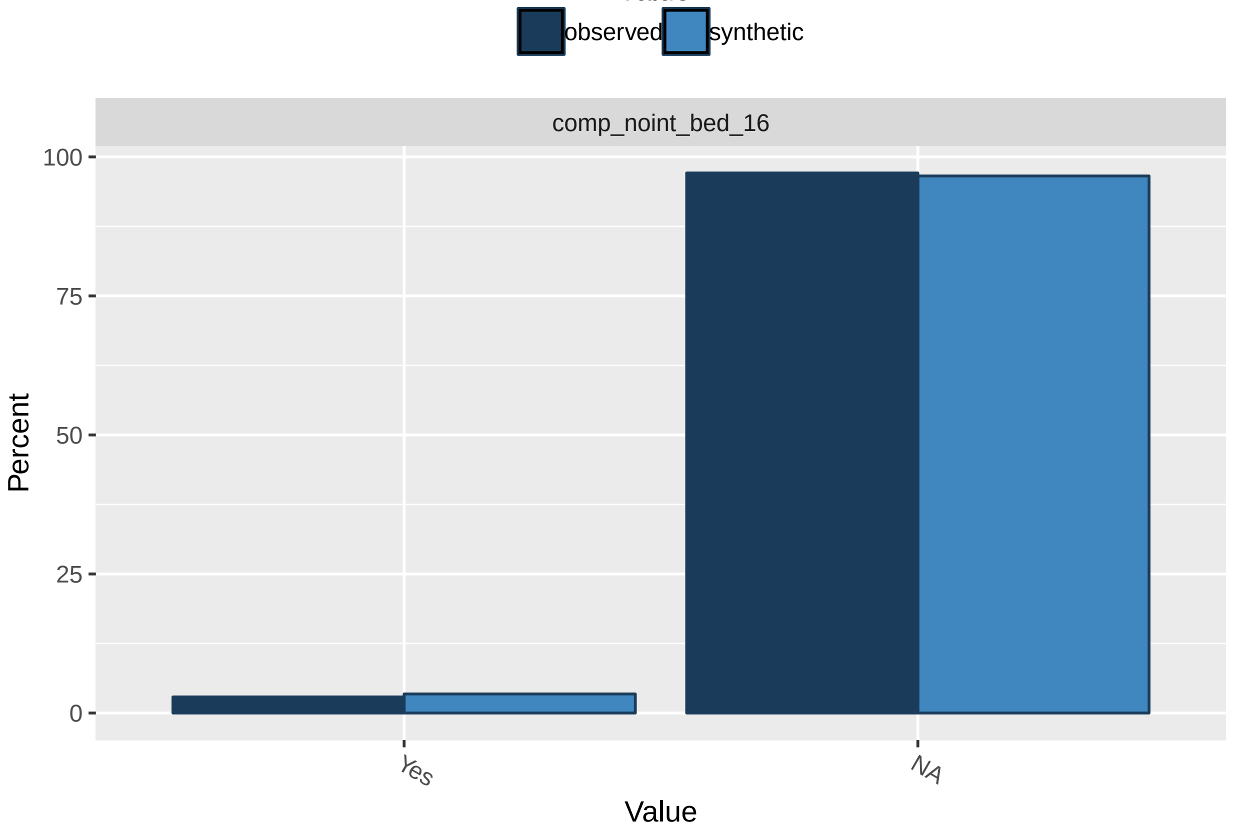

observed
synthetic
mat_dep
Percent
0
5
10
15
20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 NA
Value

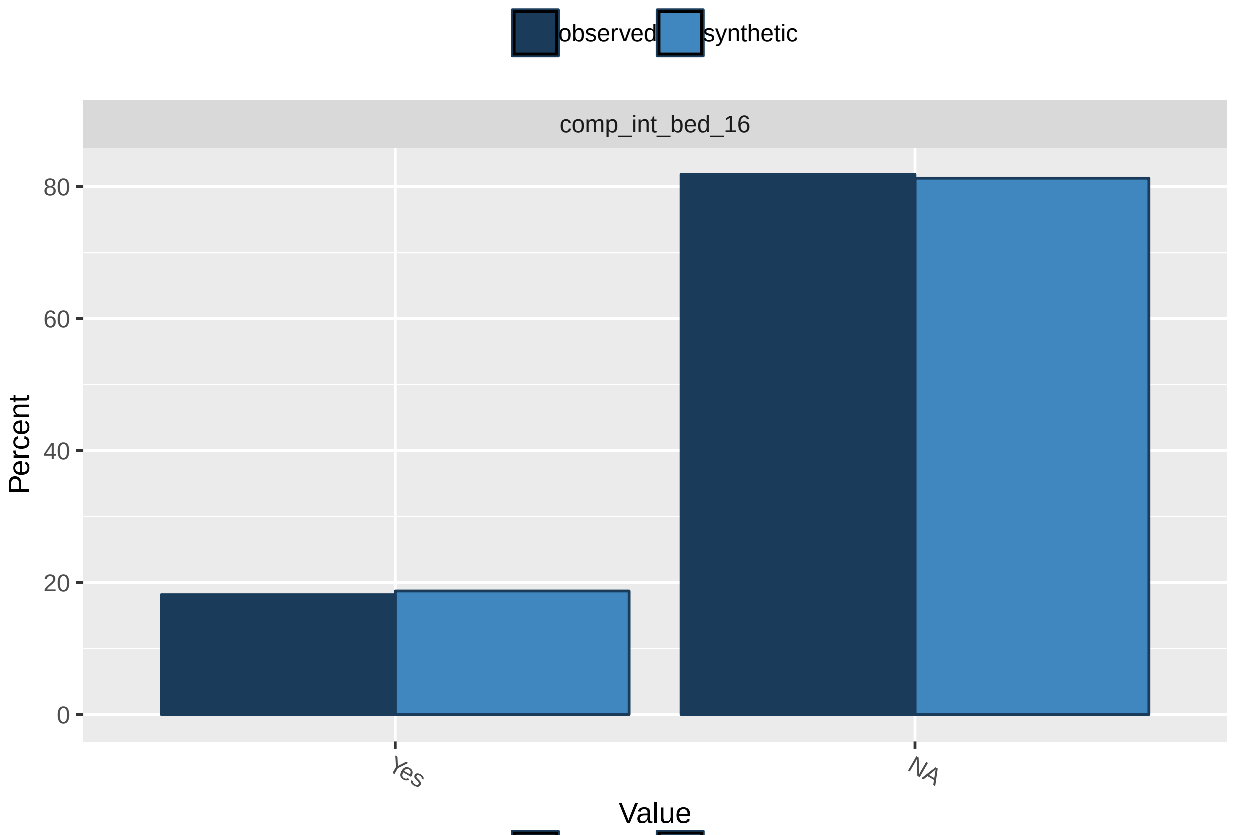

observed
synthetic
mat_age
Percent
0
2
4
6
8
<16 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 >43 NA
Value

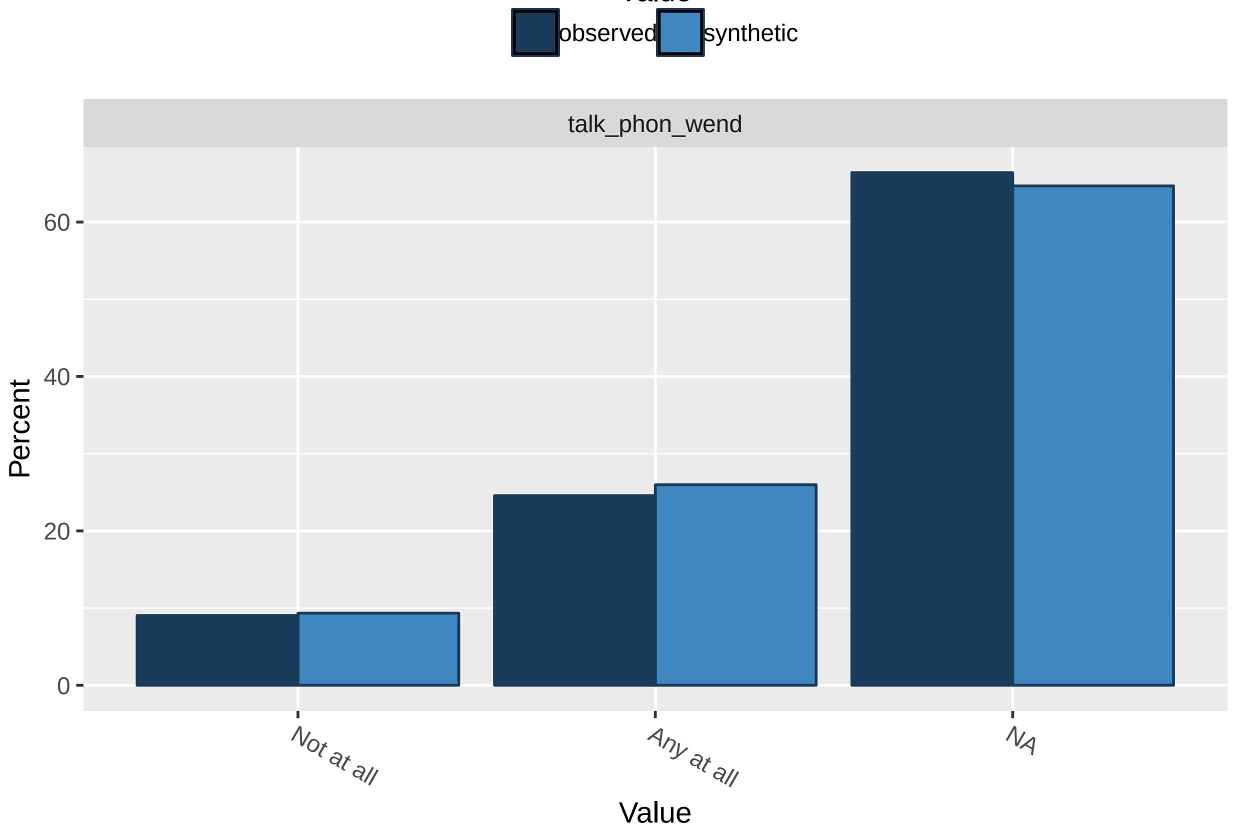

observed
synthetic
weight_16
Percent
60
40
20
0
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
miss.
Value

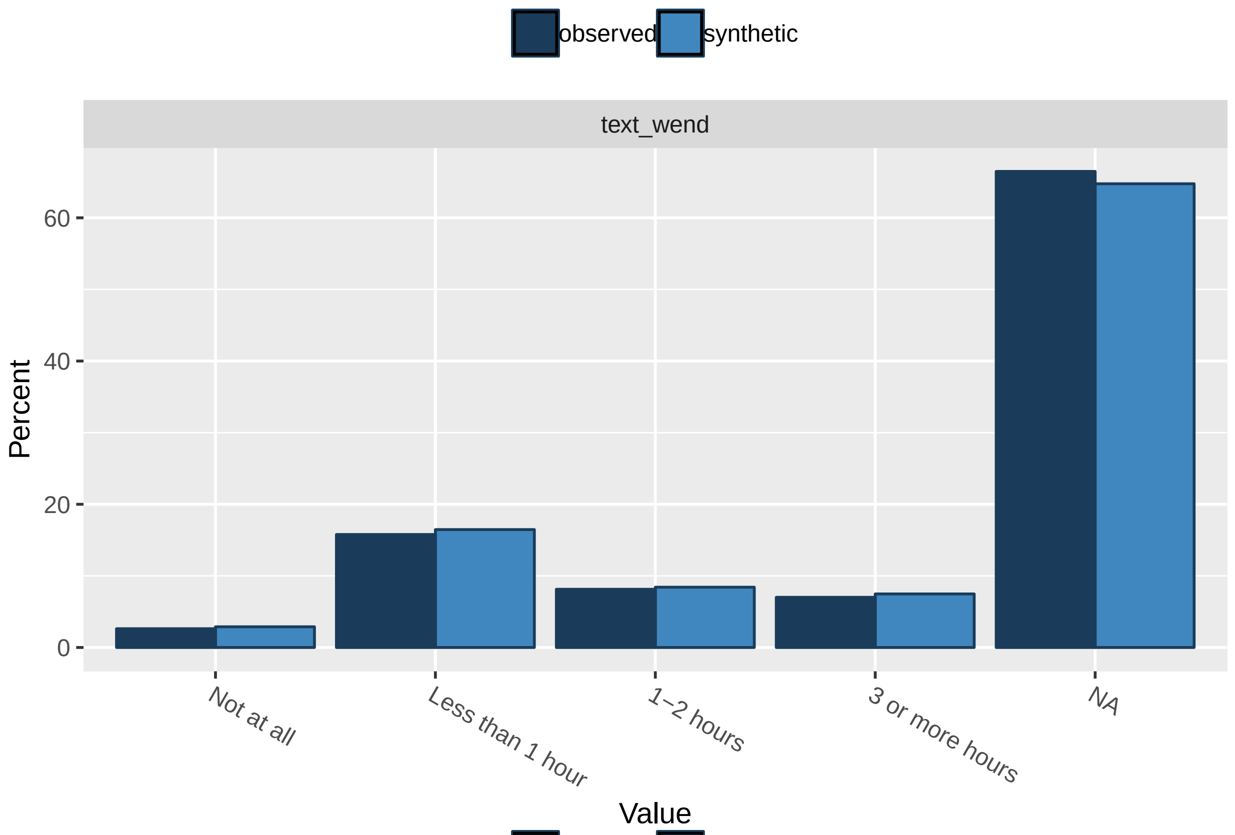

observed
synthetic
height_16
Percent
60
40
20
0
135
140
145
150
155
160
165
170
175
180
185
190
195
200
miss.N
Value

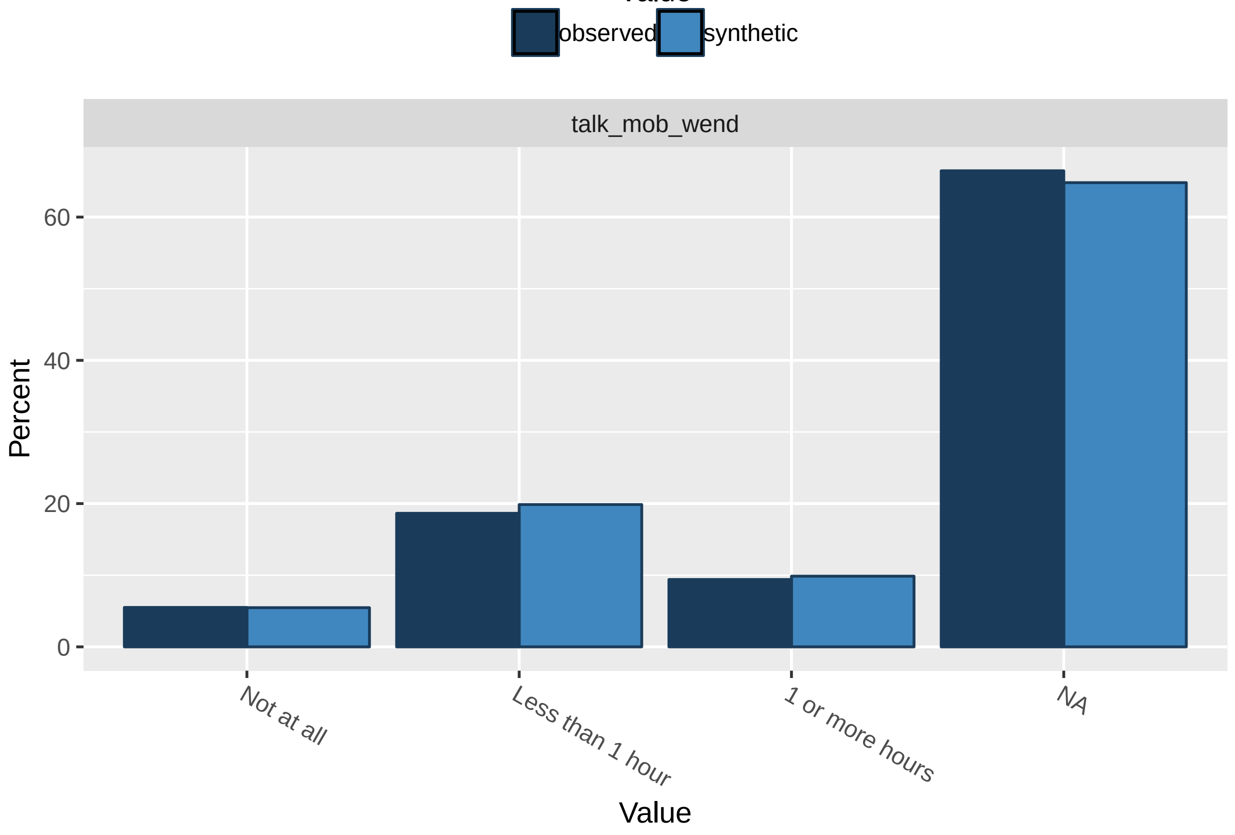

observed
synthetic
iq
Percent
50
40
30
20
10
0
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
miss.
Value

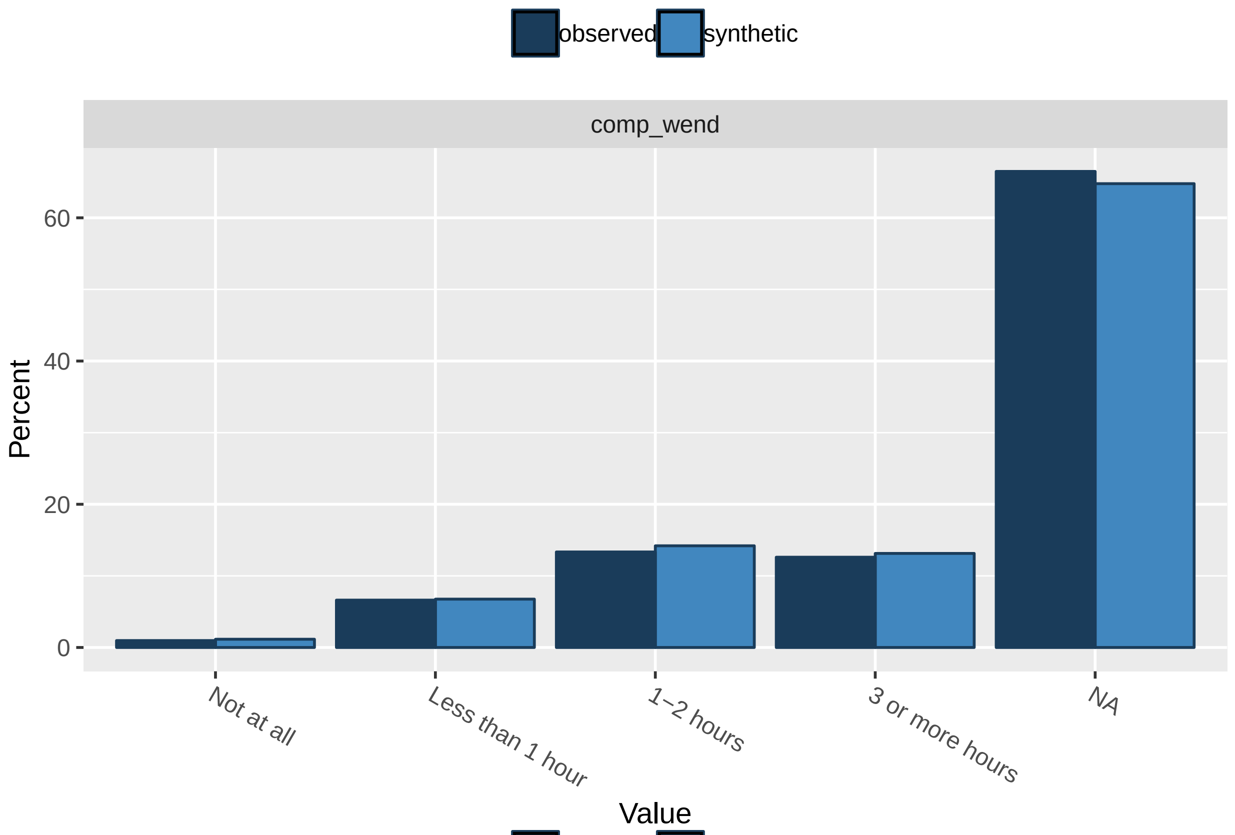

observed
synthetic
comp_noint_bed_16
Percent
100
75
50
25
0
Yes
NA
Value

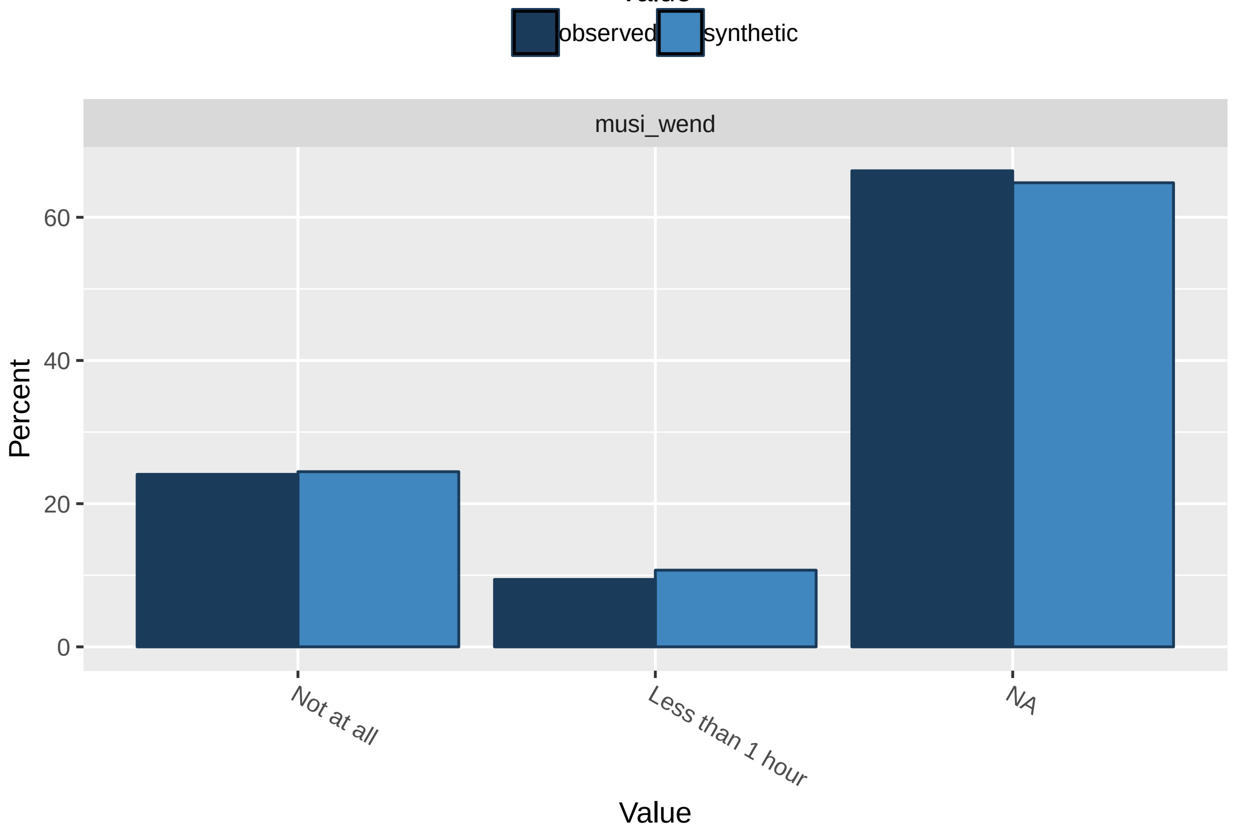

observed
synthetic
comp_int_bed_16
Percent
80
60
40
20
0
Yes
NA
Value

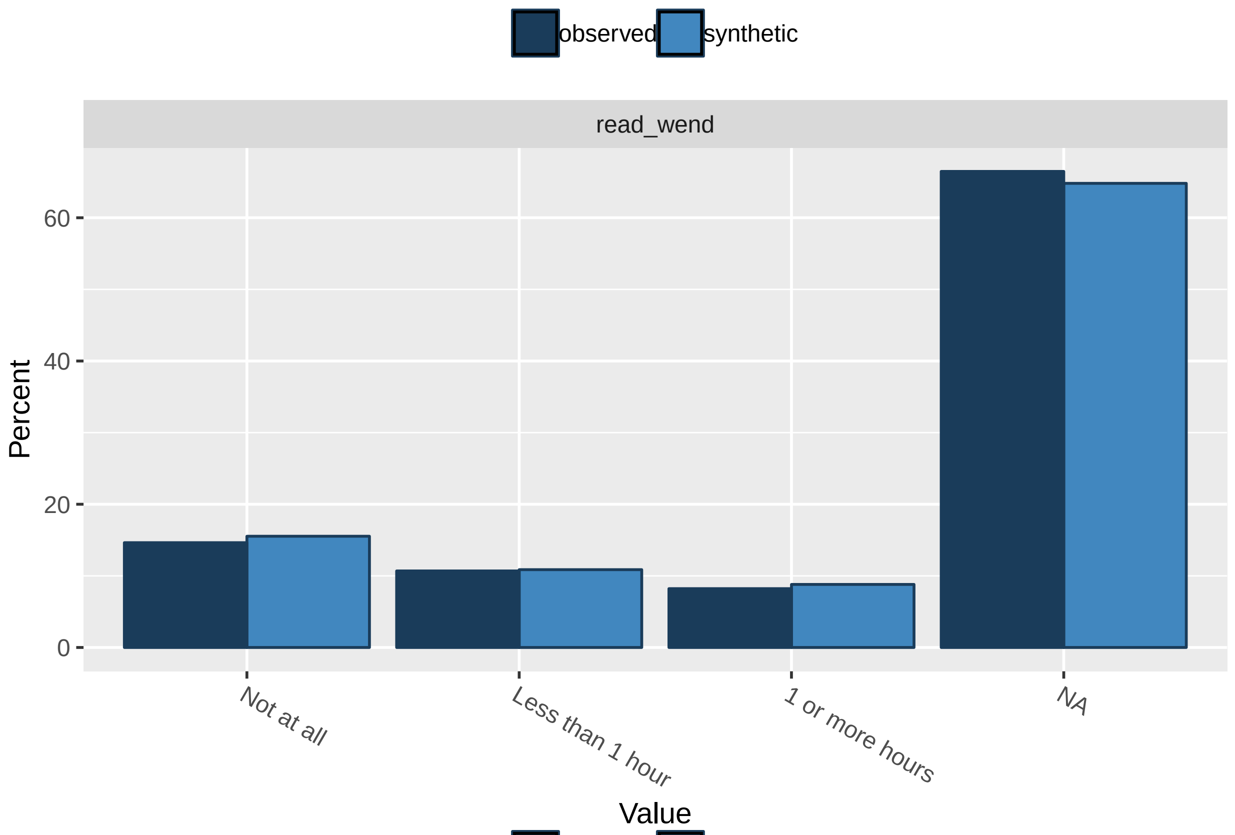

observed
synthetic
talk_phon_wend
Percent
60
40
20
0
Not at all
Any at all
NA
Value

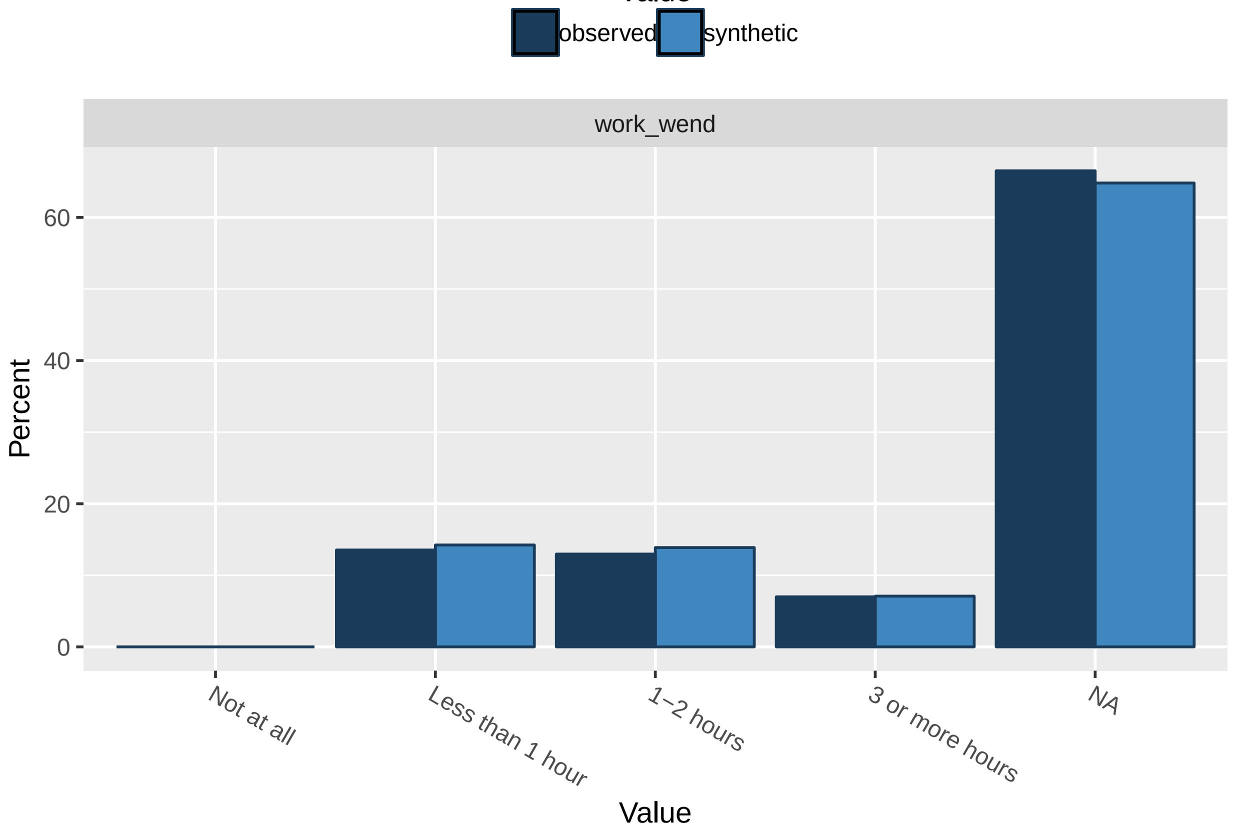

observed
synthetic
text_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

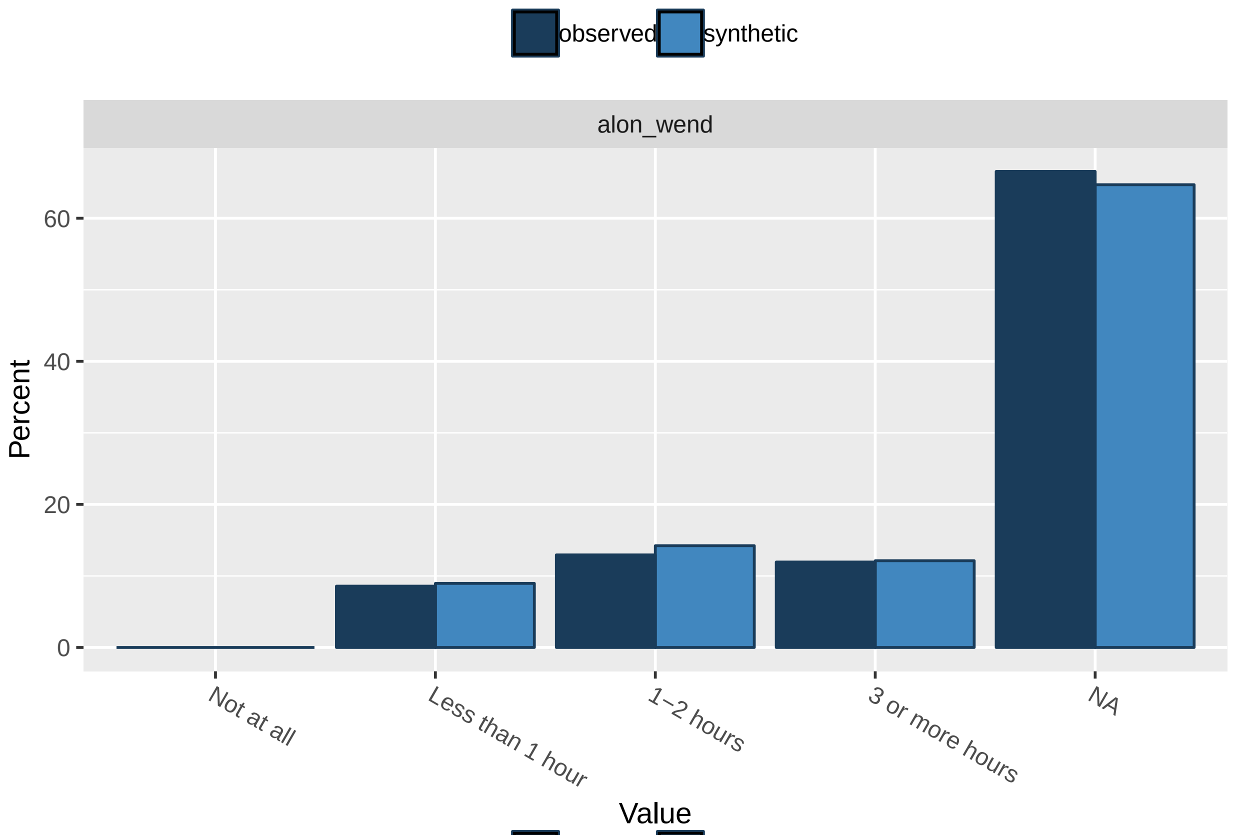

observed
synthetic
talk_mob_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
1 or more hours
NA
Value


observed
synthetic
comp_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

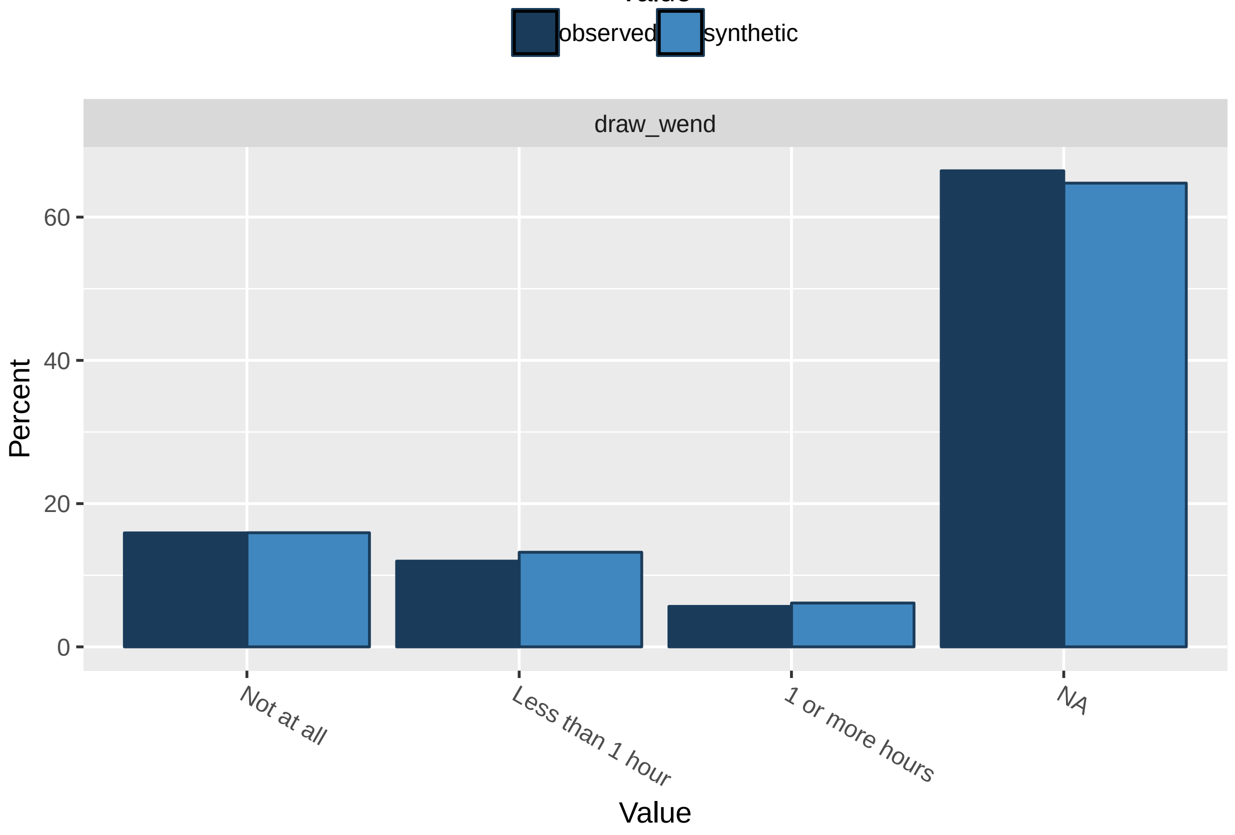

observed
synthetic
musi_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
NA
Value


observed
synthetic
read_wend
Percent
60
40
20
0
Not at all
Less than 1 hour
1 or more hours
NA
Value

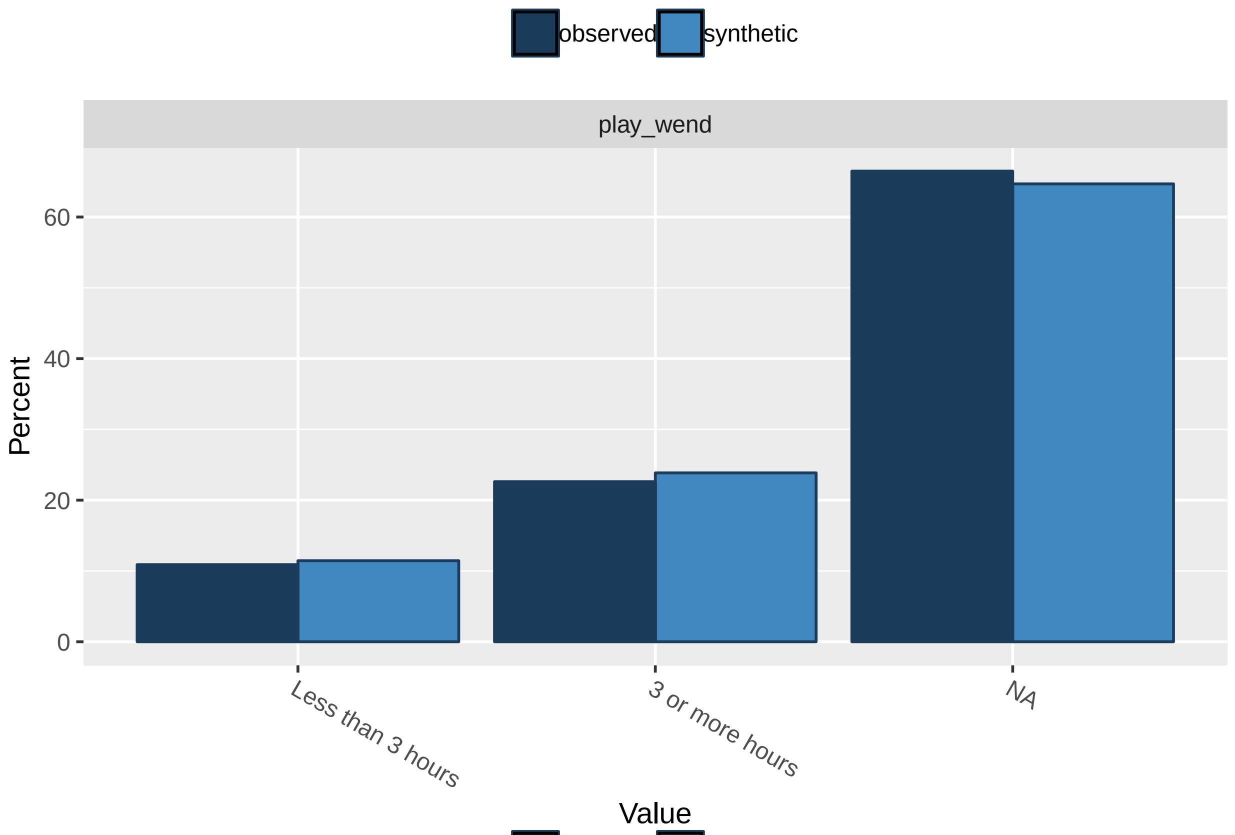

observed
synthetic
work_wend
Percent
60
40
20
0
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

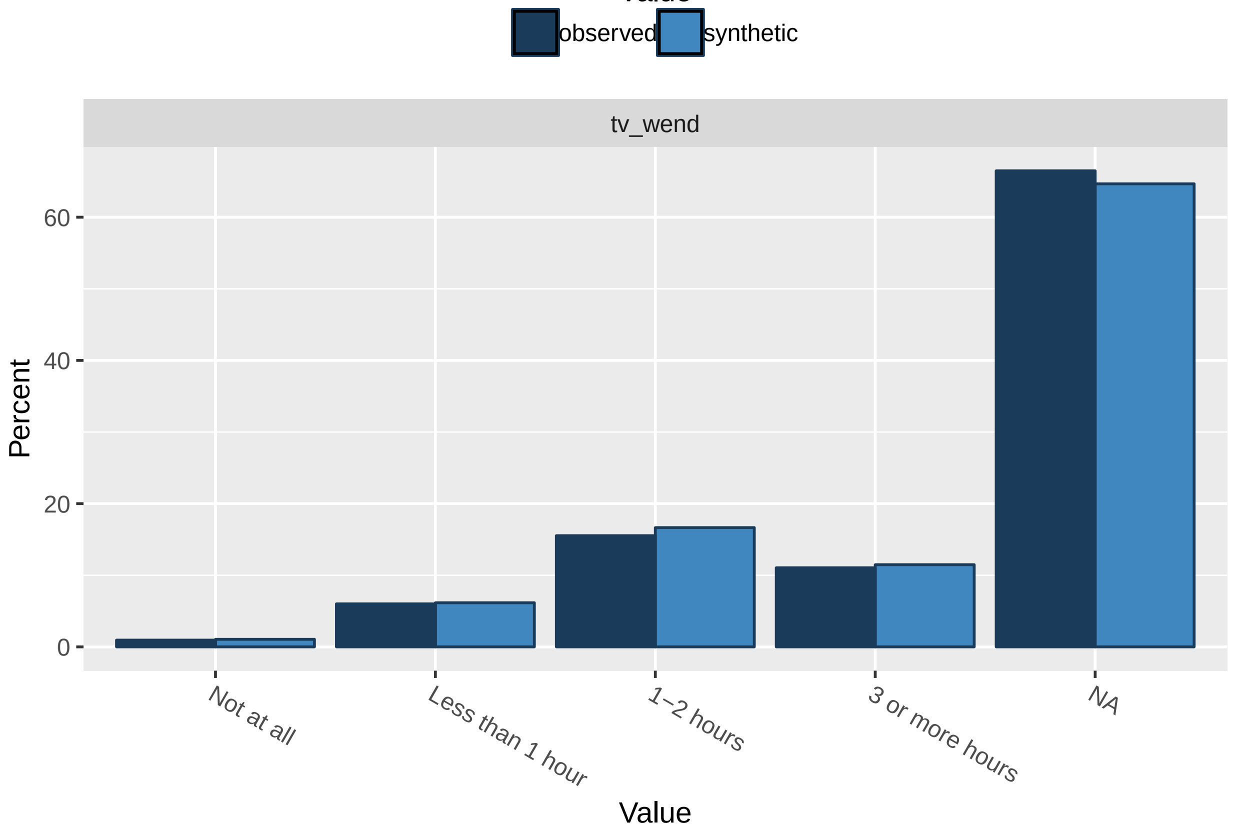


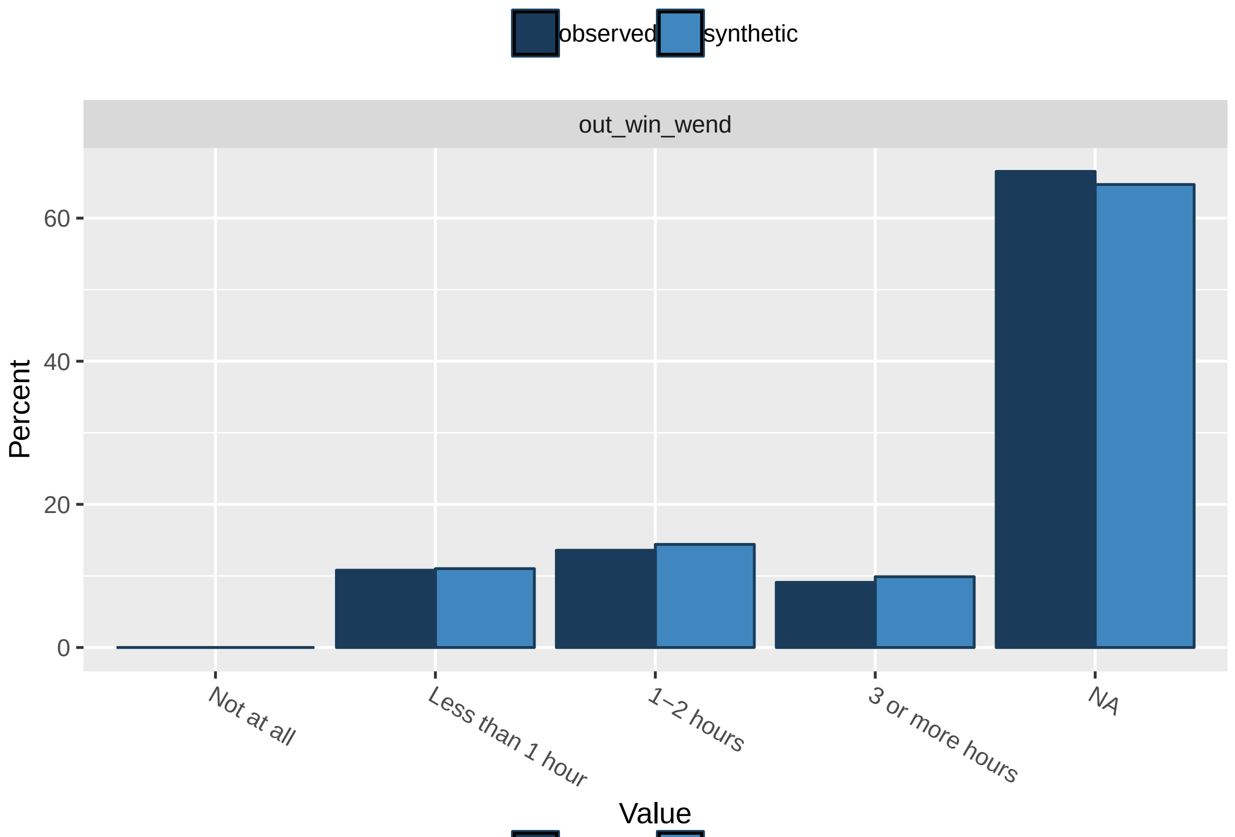
observed
synthetic
alon_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

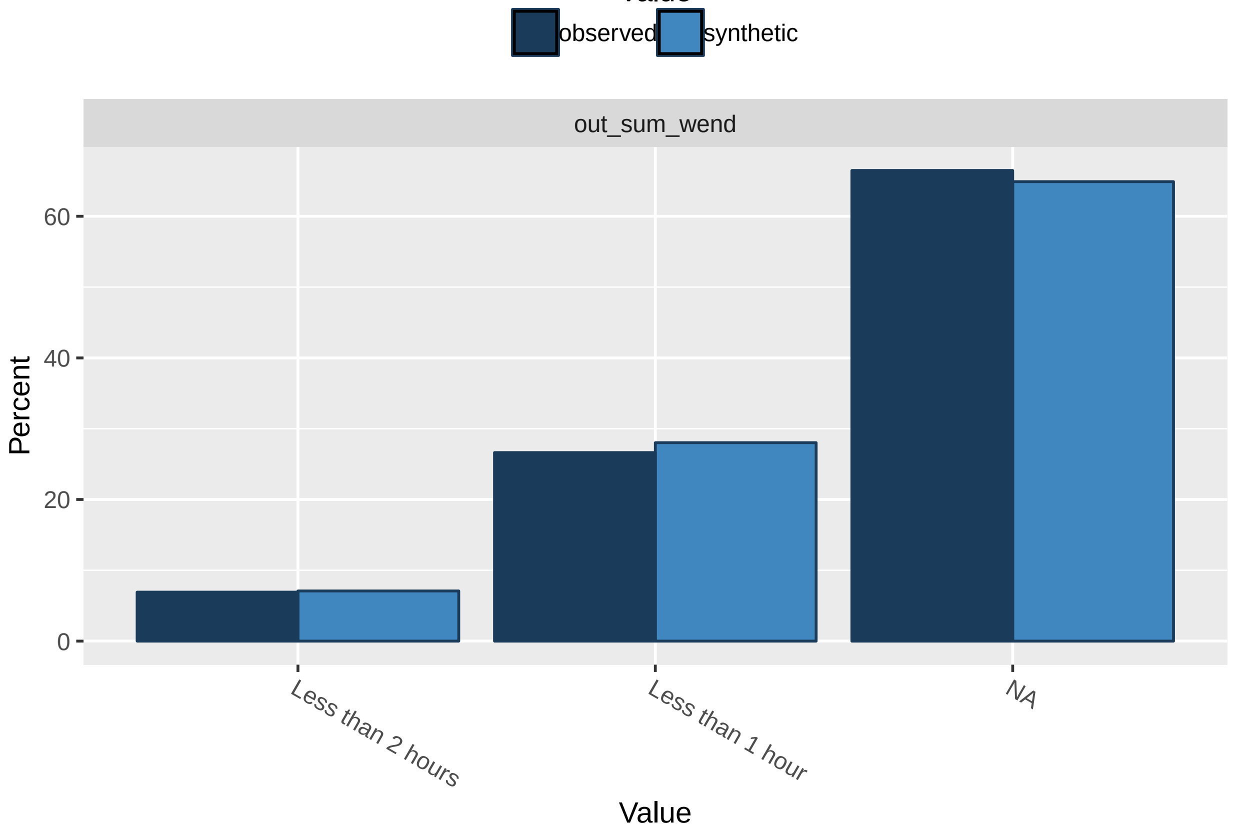


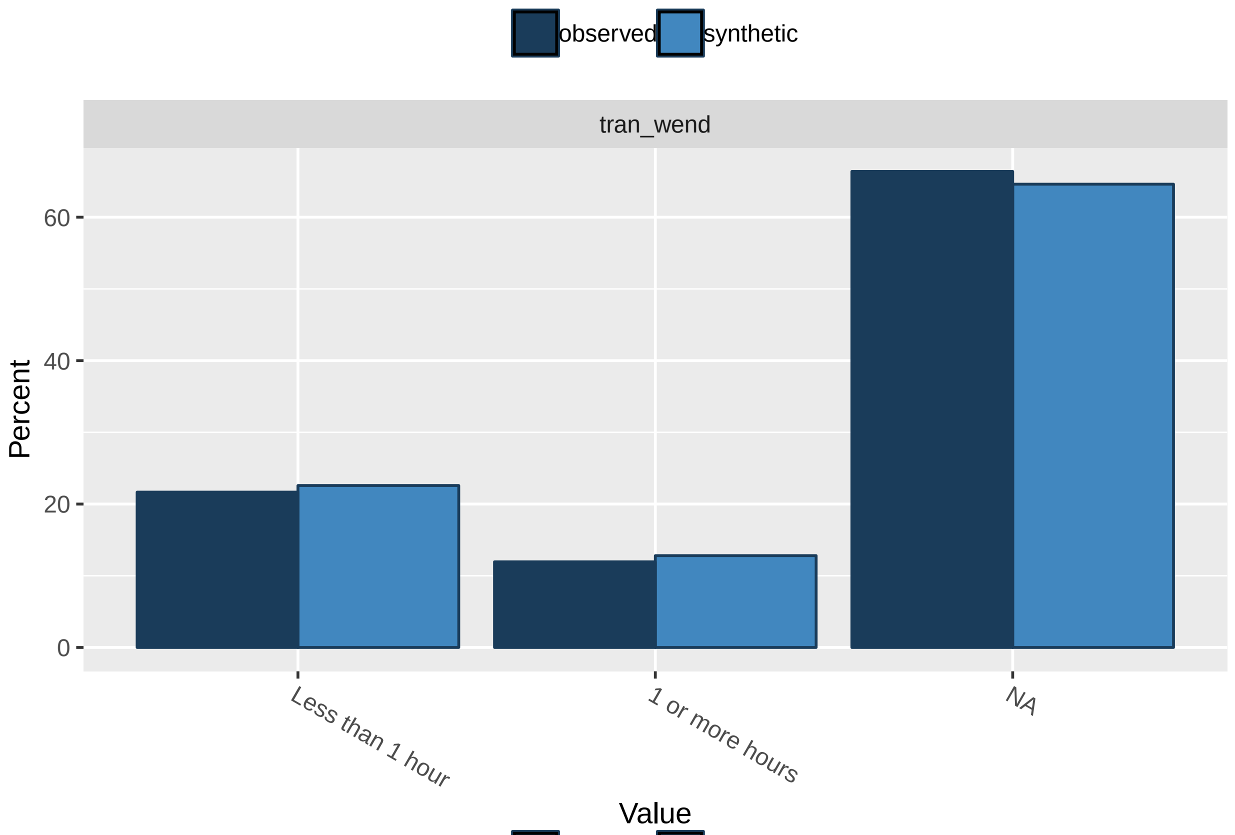
observed
synthetic
draw_wend
Percent
0
20
40
60
Not at all
Less than 1 hour
1 or more hours
NA
Value

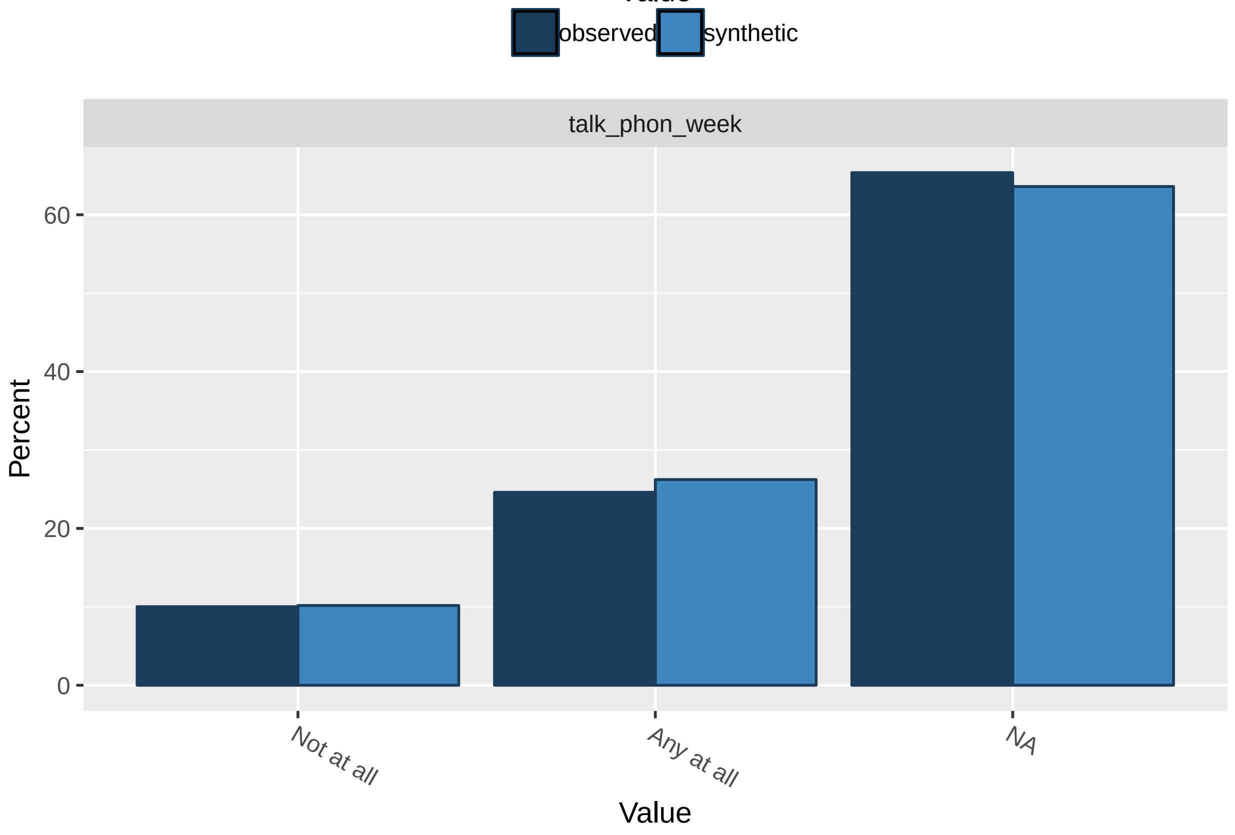

observed
synthetic
play_wend
Percent
60
40
20
0
Less than 3 hours
3 or more hours
NA
Value

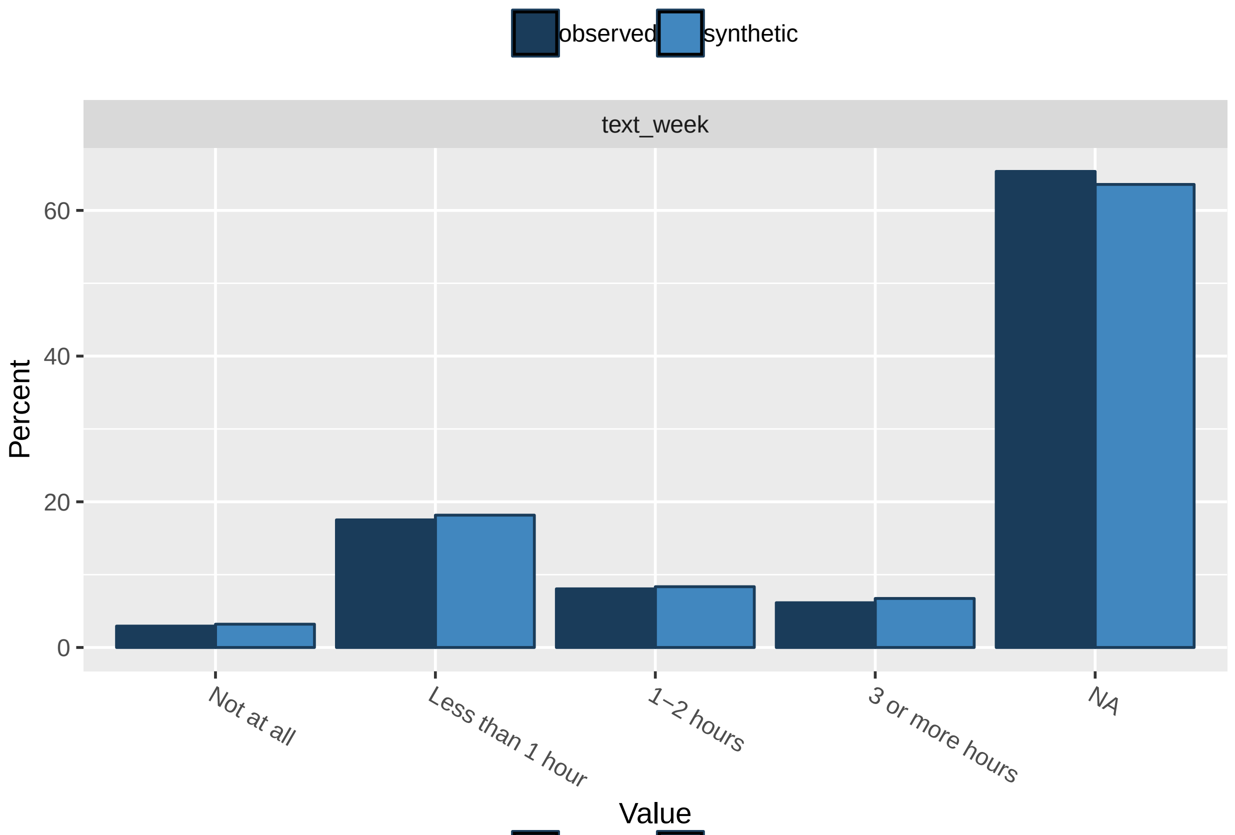

observed
synthetic
tv_wend
Percent
60
40
20
0
Not at all
Less than 1 hour
1−2 hours
3 or more hours
NA
Value

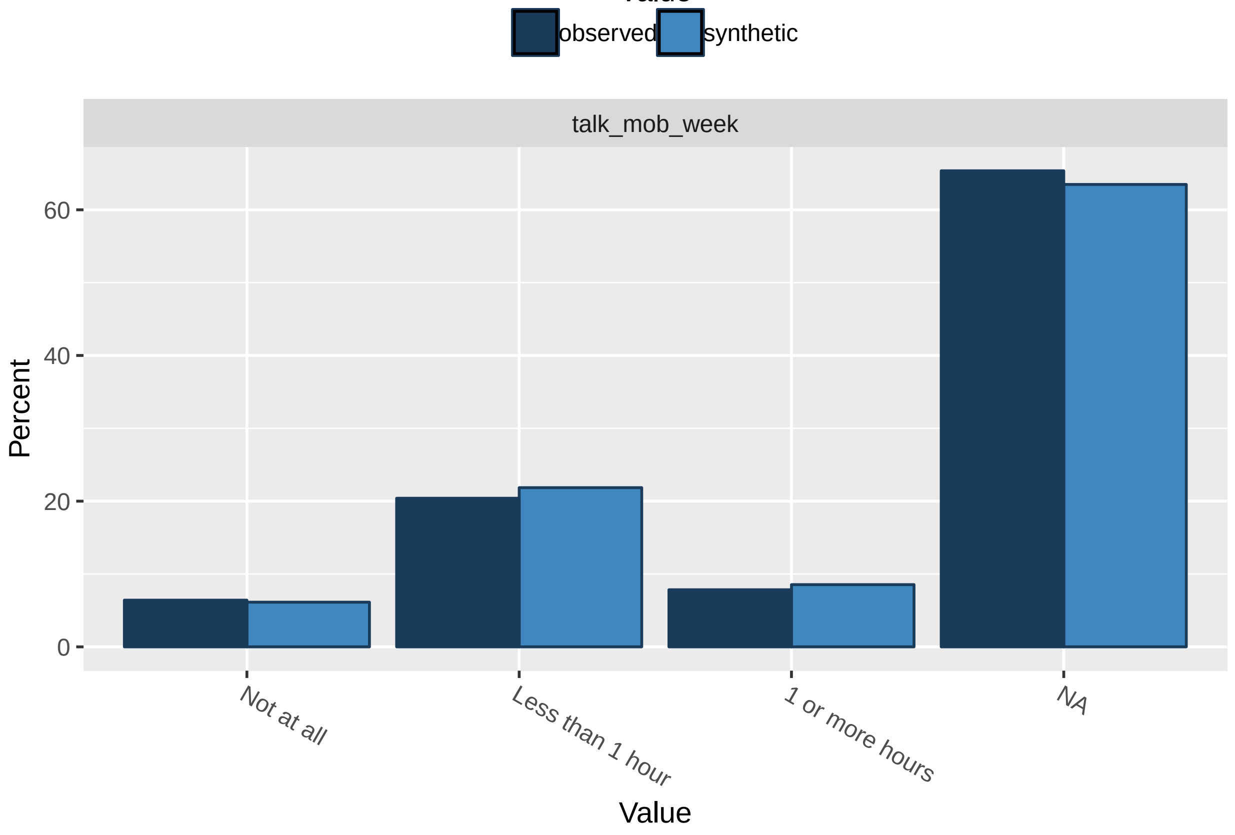

observed
synthetic
out_win_wend
Percent
60
40
20
0
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

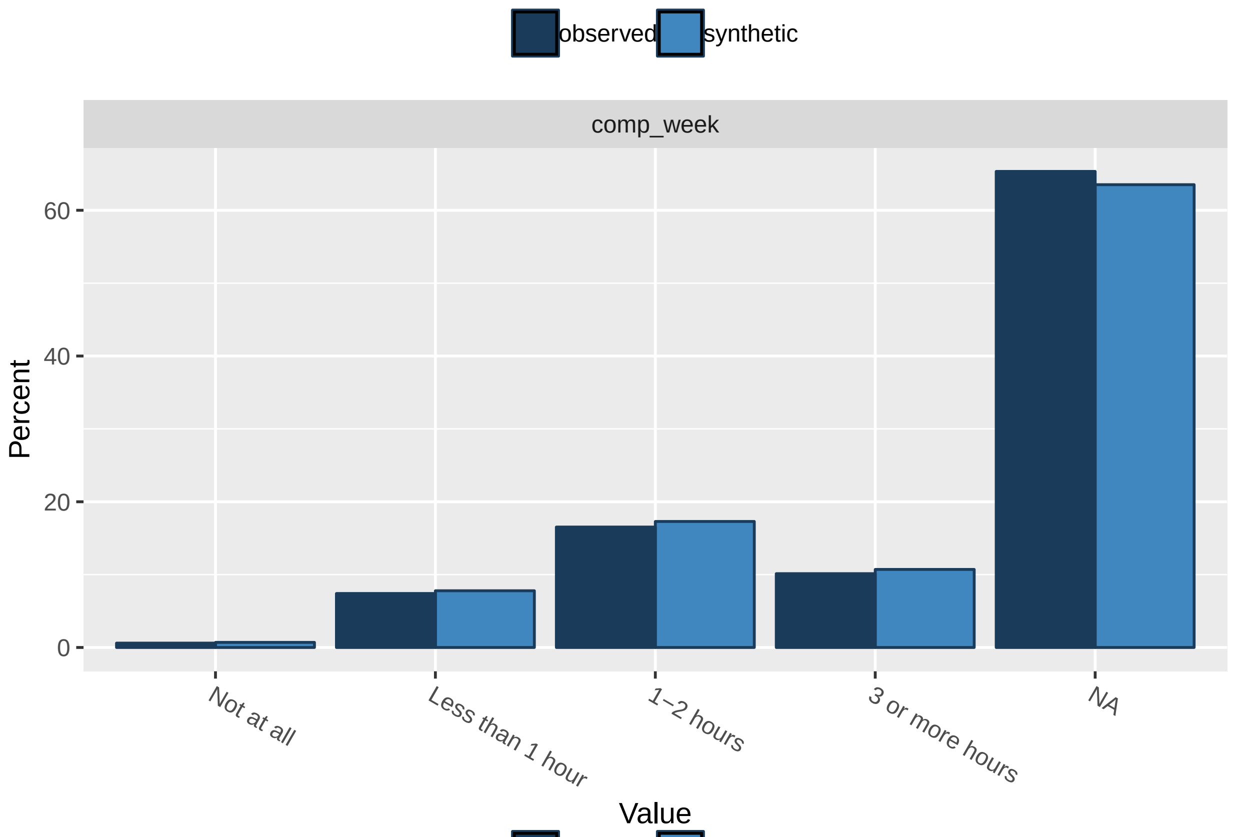

observed
synthetic
out_sum_wend
Percent
60
40
20
0
Less than 2 hours
Less than 1 hour
NA
Value

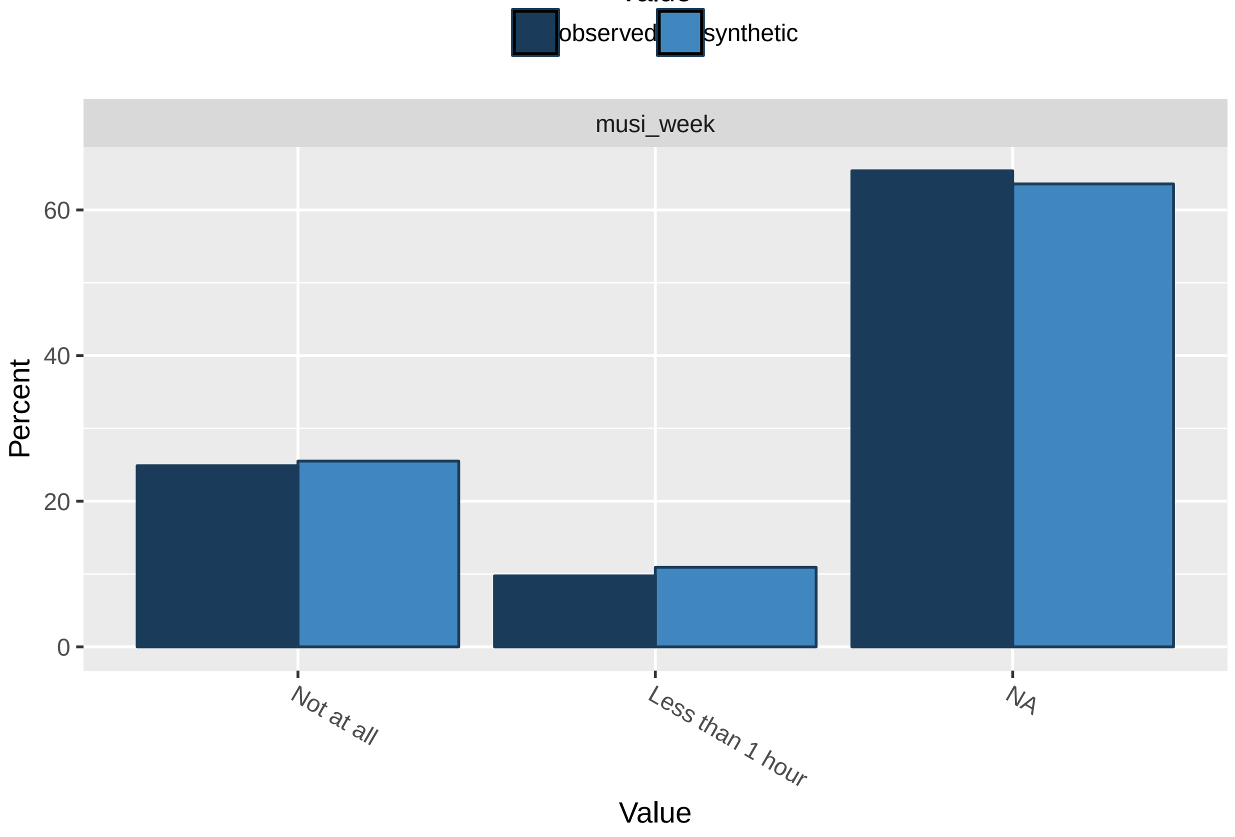

observed
synthetic
tran_wend
Percent
0
20
40
60
Less than 1 hour
1 or more hours
NA
Value

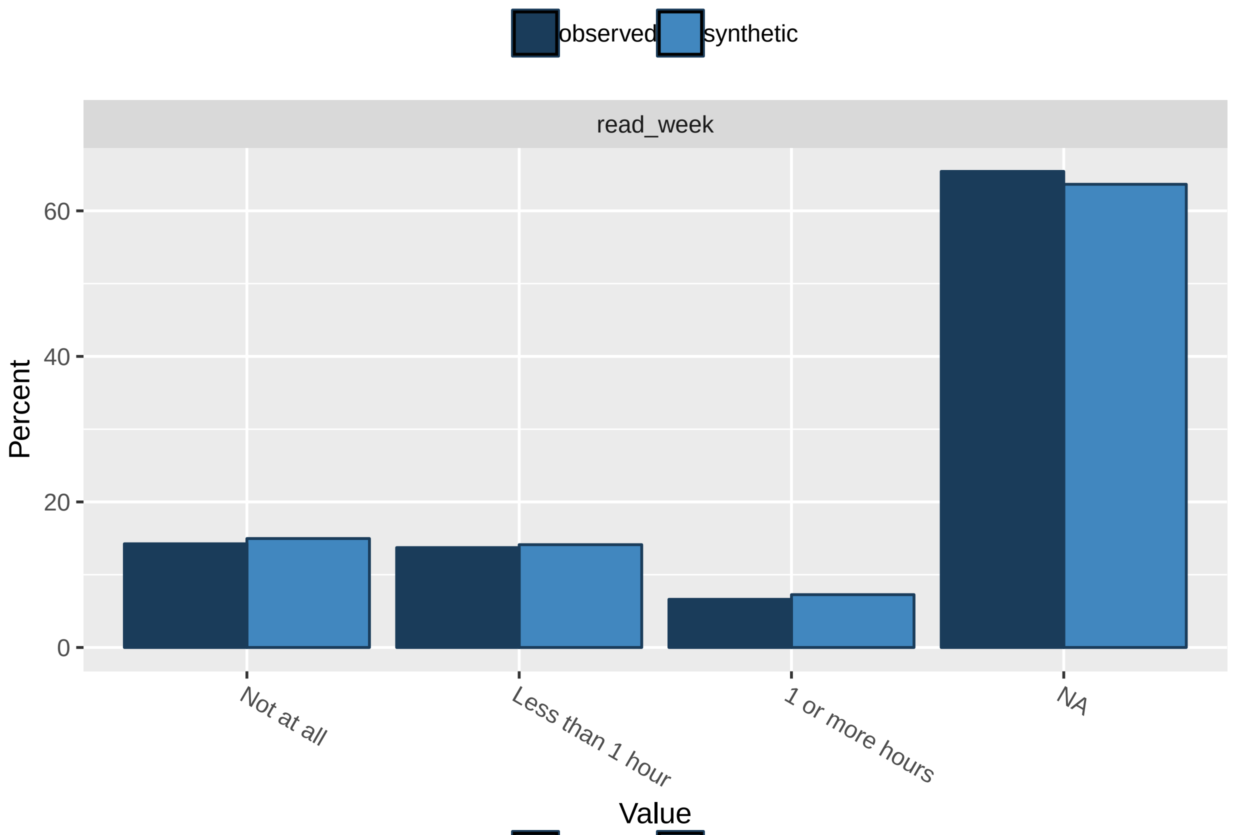

observed
synthetic
talk_phon_week
Percent
0
20
40
60
Not at all
Any at all
NA
Value

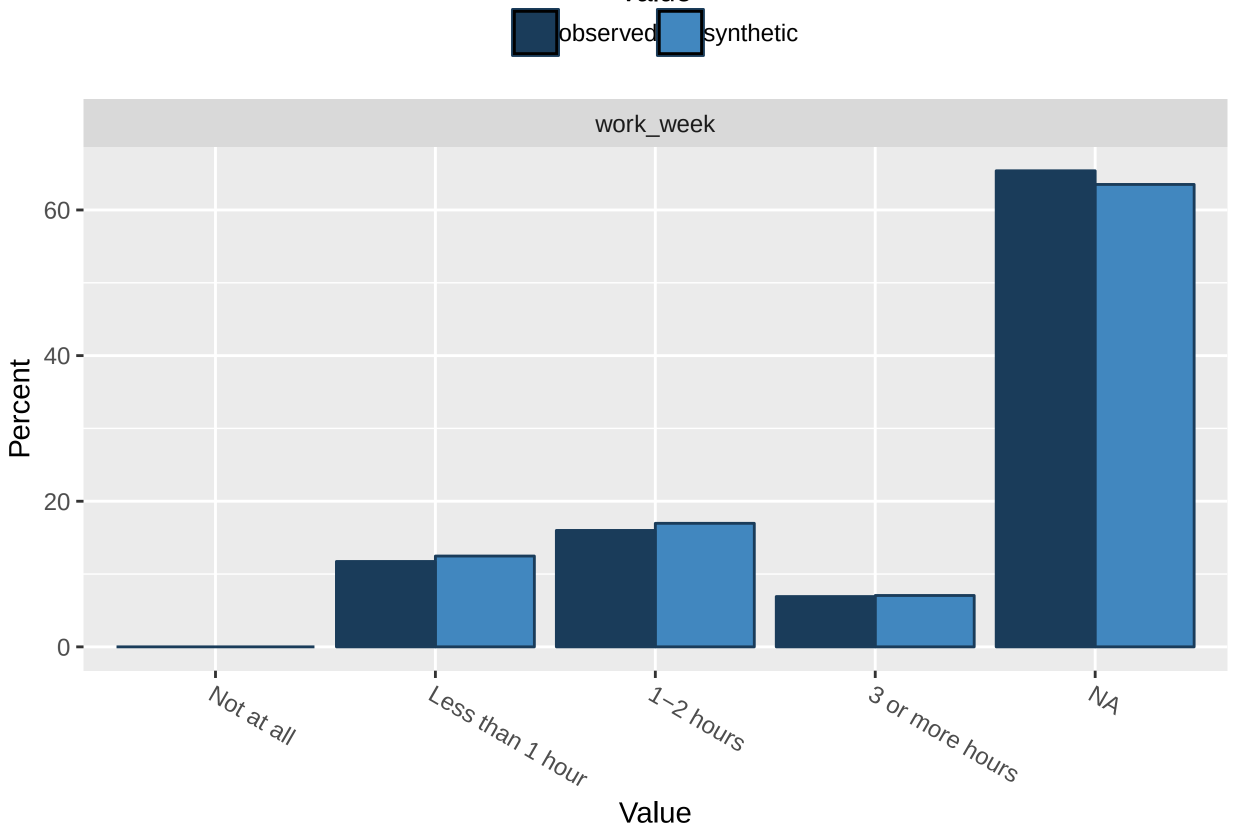

observed
synthetic
text_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

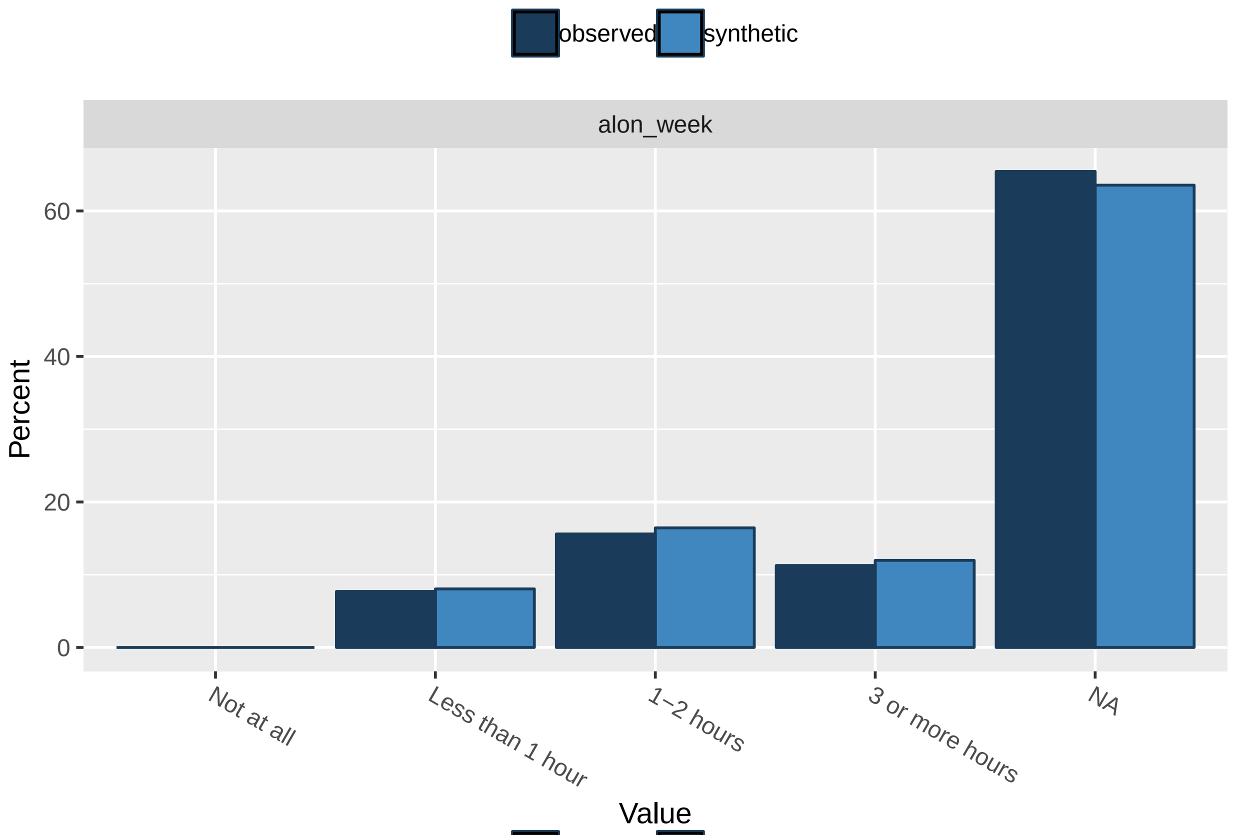

observed
synthetic
talk_mob_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1 or more hours
NA
Value

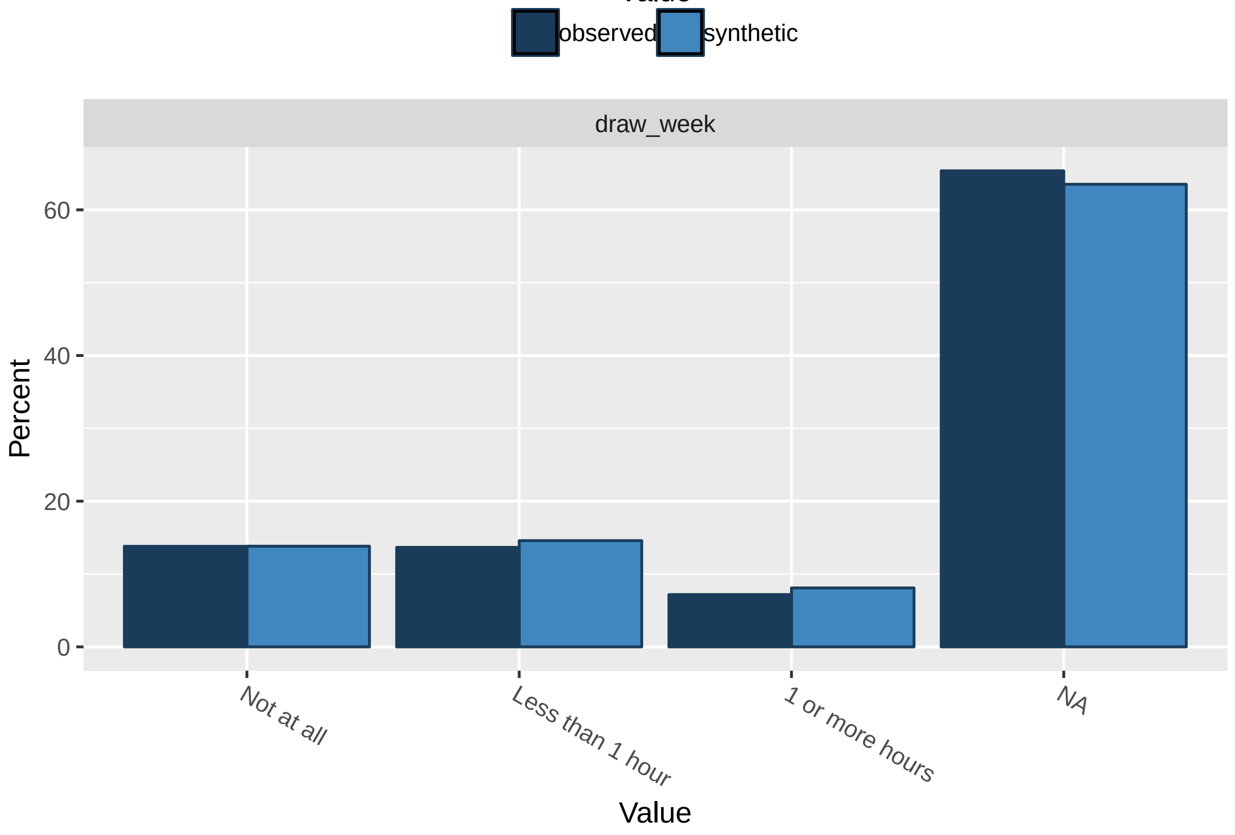

observed
synthetic
comp_week
Percent
60
40
20
0
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

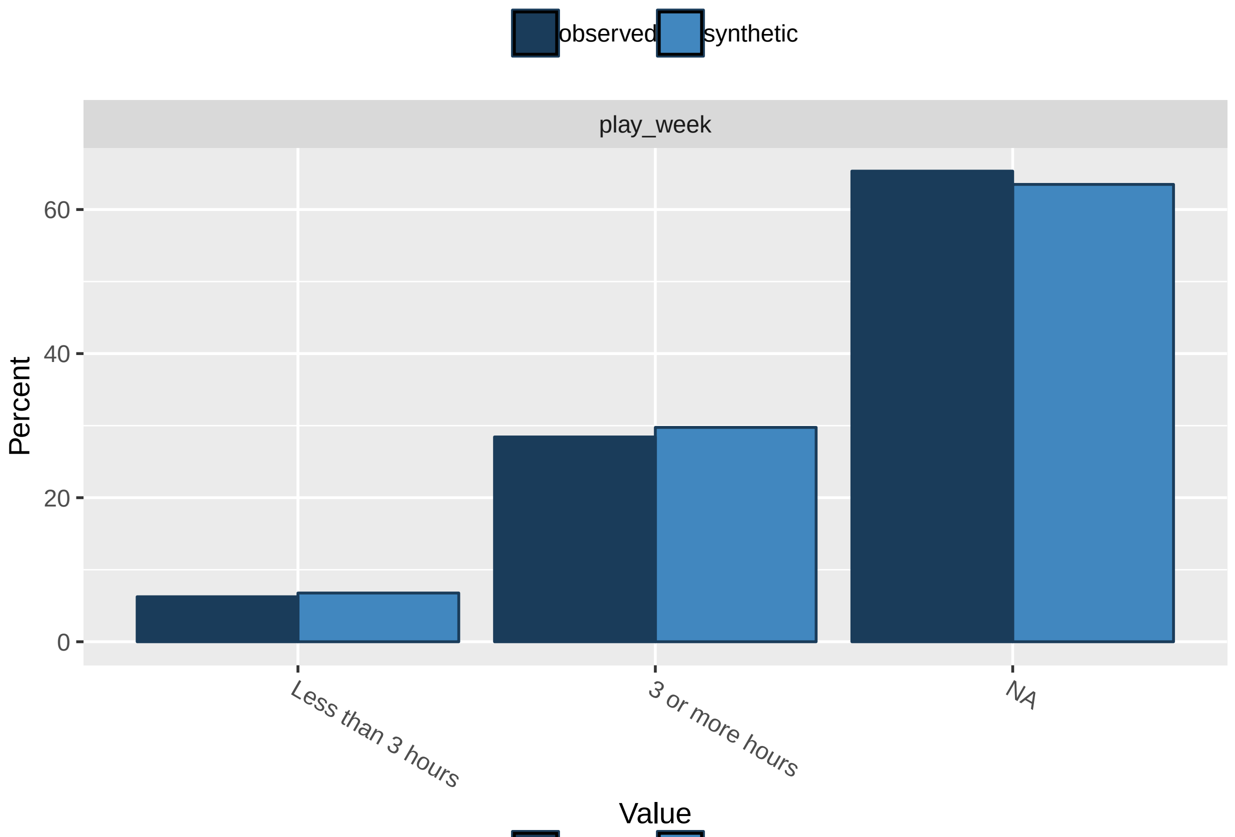

observed
synthetic
musi_week
Percent
60
40
20
0
Not at all
Less than 1 hour
NA
Value

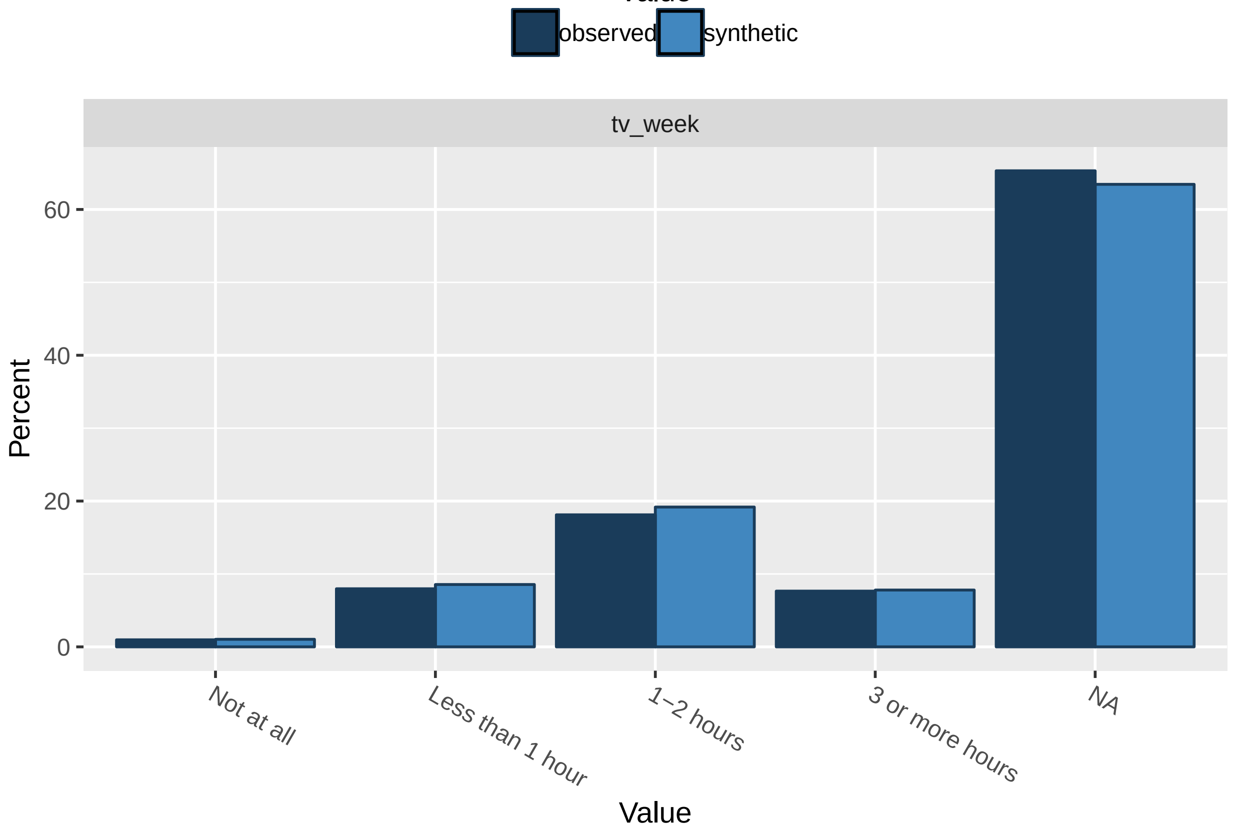

observed
synthetic
read_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1 or more hours
NA
Value

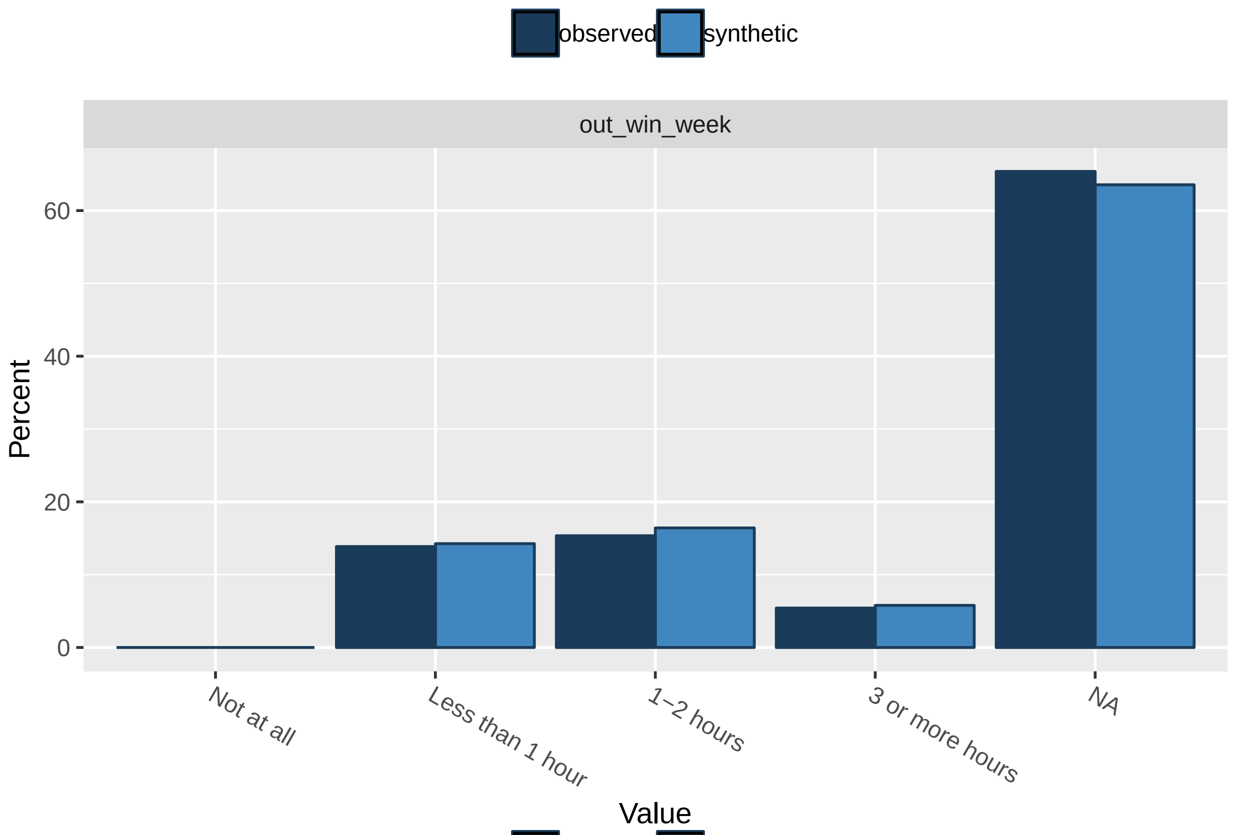

observed
synthetic
work_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

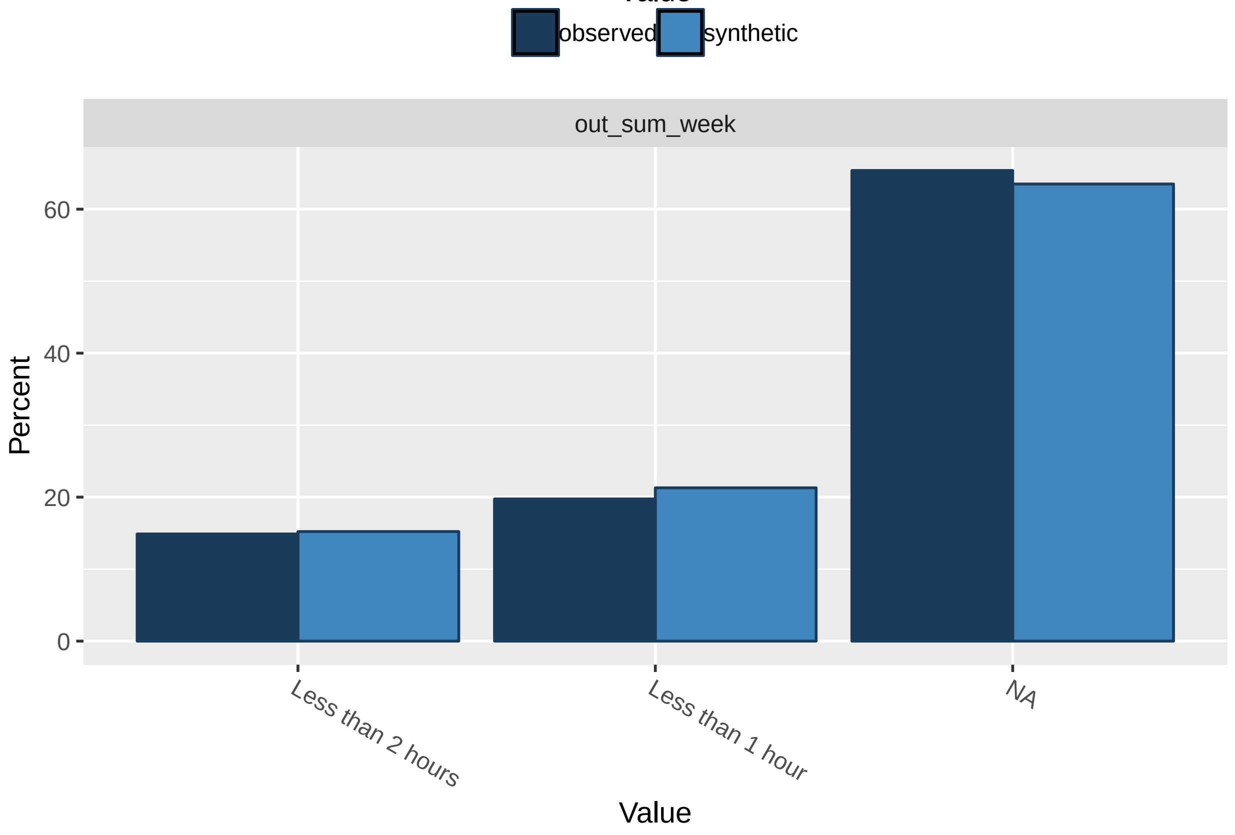

observed
synthetic
alon_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

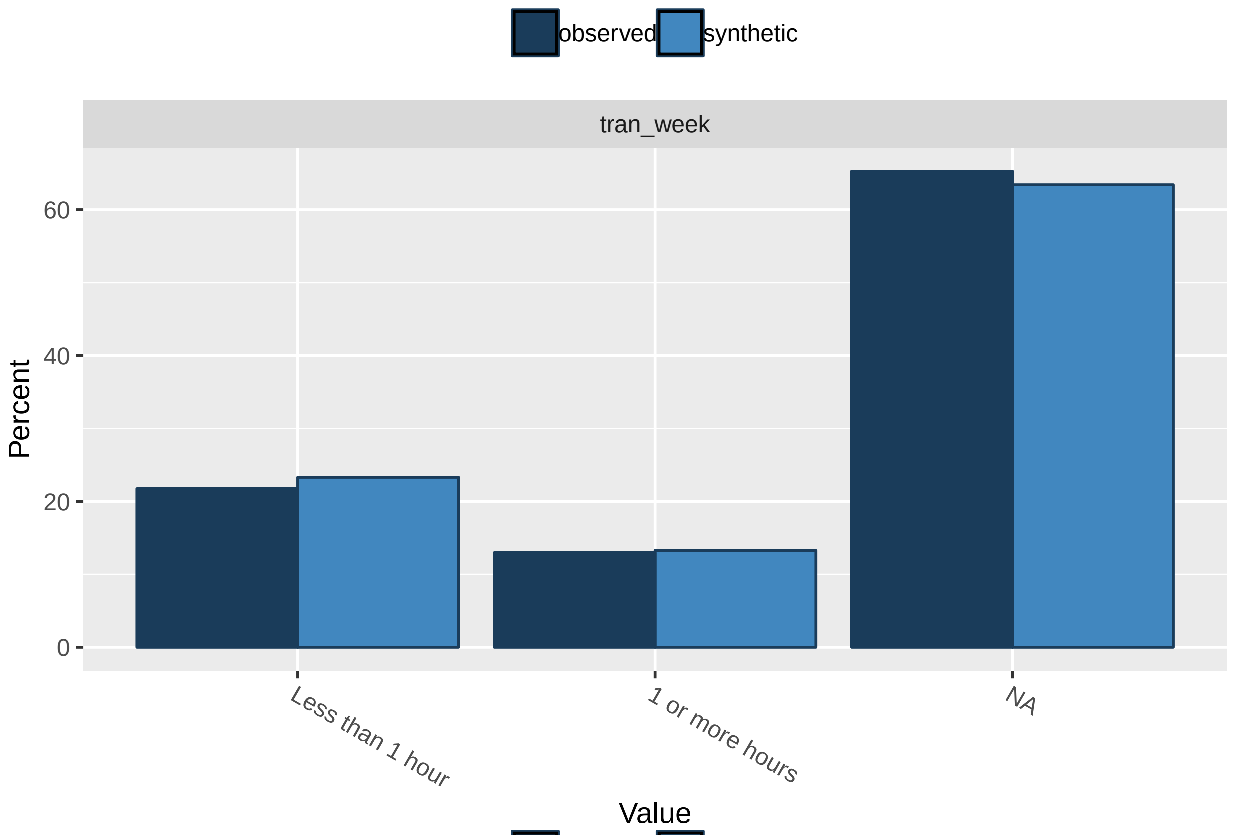

observed
synthetic
draw_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1 or more hours
NA
Value


observed
synthetic
play_week
Percent
0
20
40
60
Less than 3 hours
3 or more hours
NA
Value

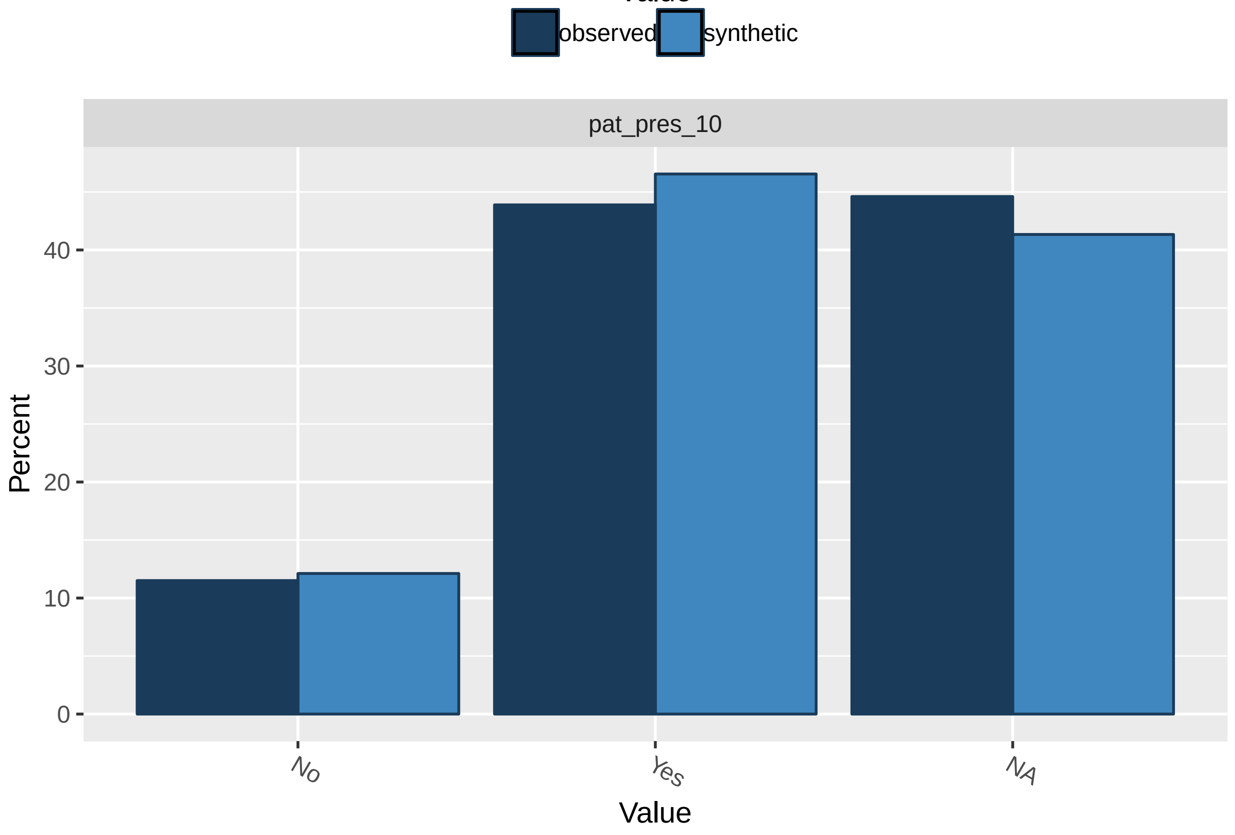

observed
synthetic
tv_week
Percent
0
20
40
60
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value


observed
synthetic
out_win_week
Percent
60
40
20
0
Not at all
Less than 1 hour
1–2 hours
3 or more hours
NA
Value

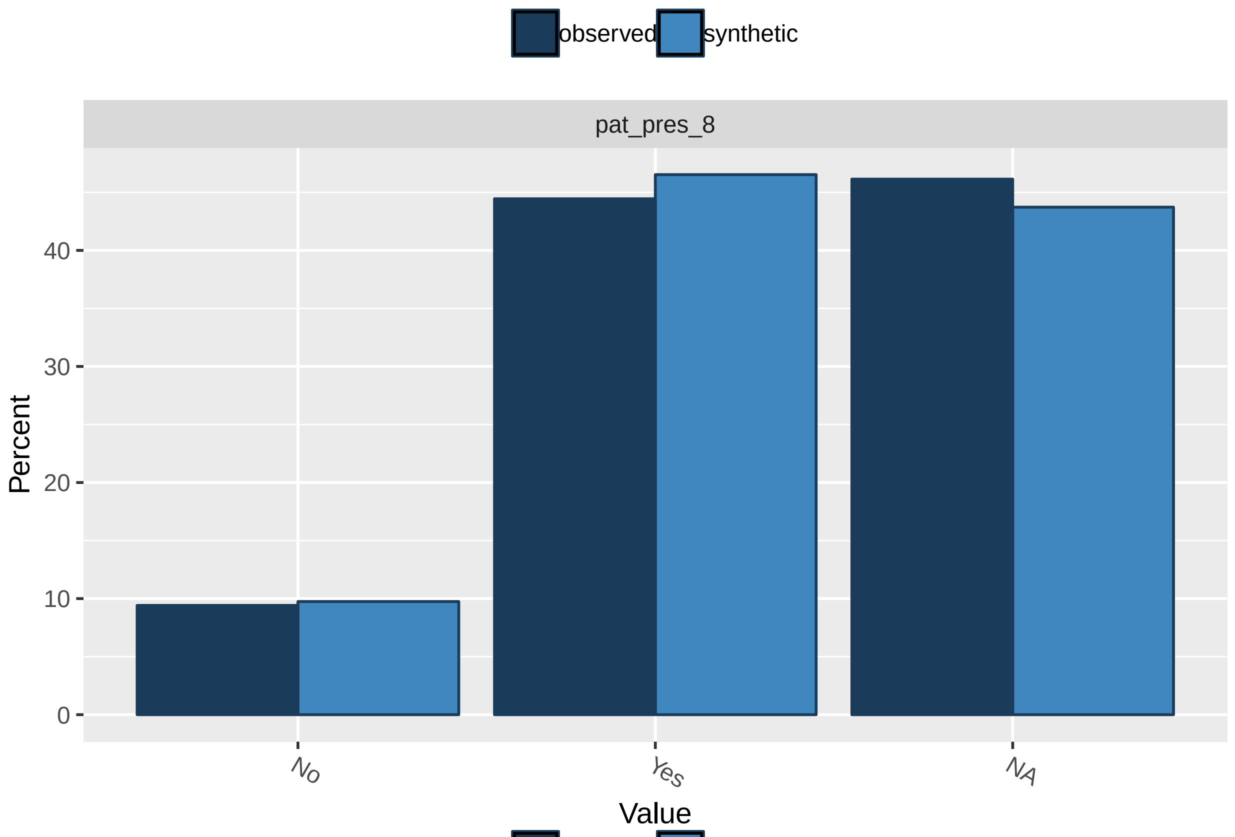

observed
synthetic
out_sum_week
Percent
60
40
20
0
Less than 2 hours
Less than 1 hour
NA
Value

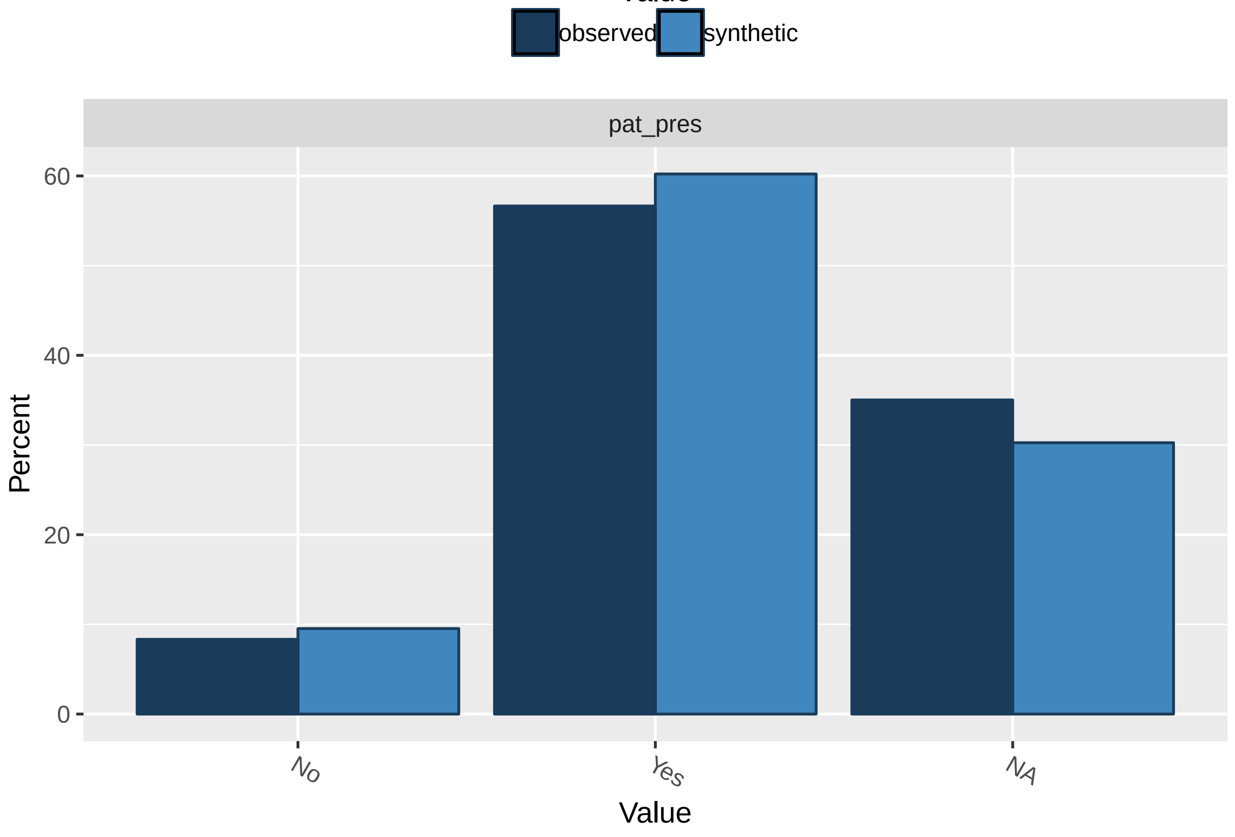


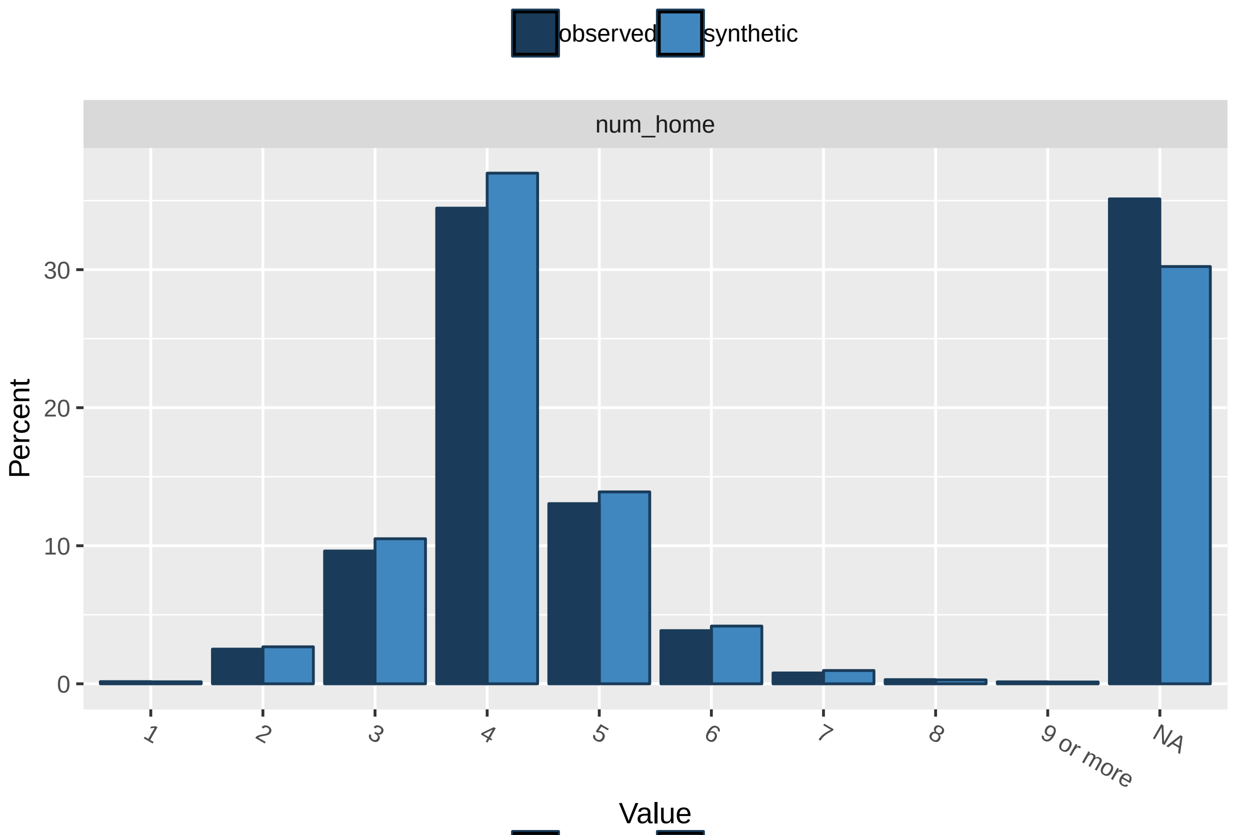
observed
synthetic
tran_week
Percent
0
20
40
60
Less than 1 hour
1 or more hours
NA
Value

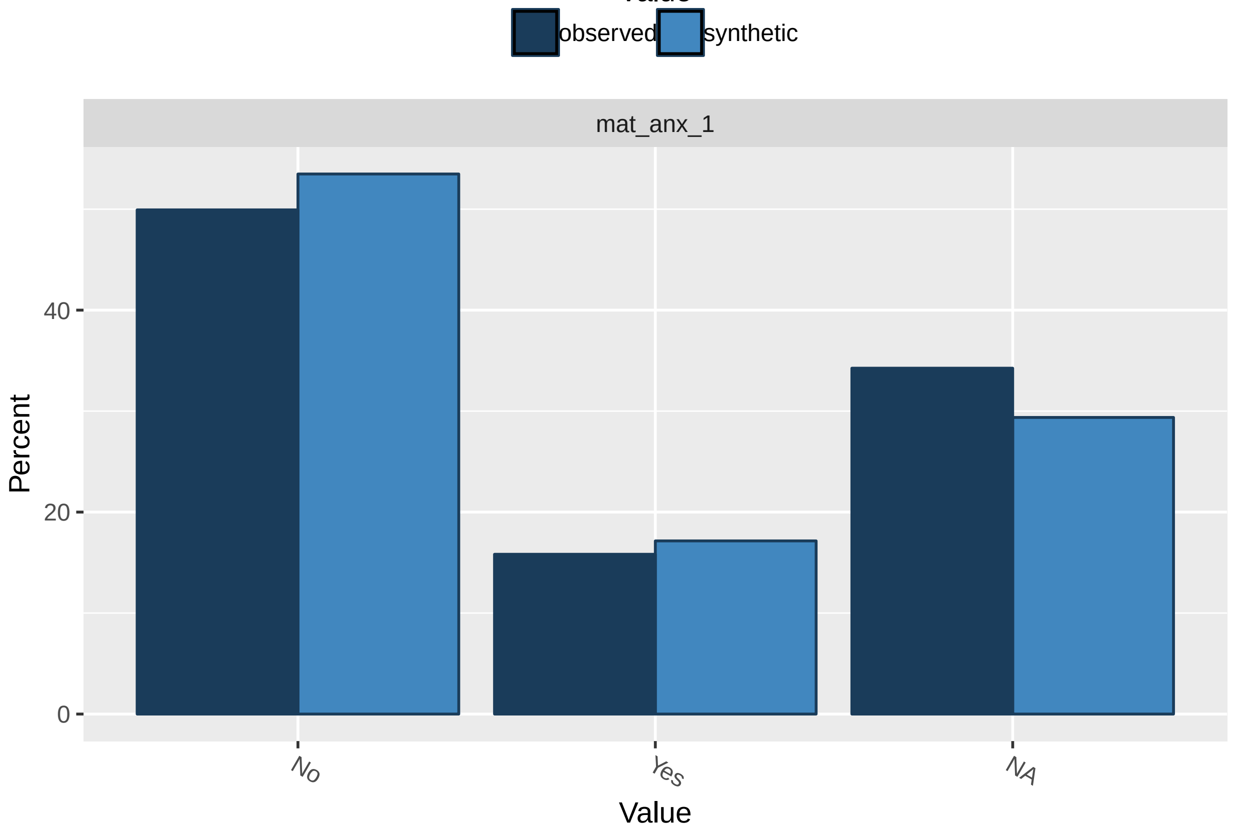


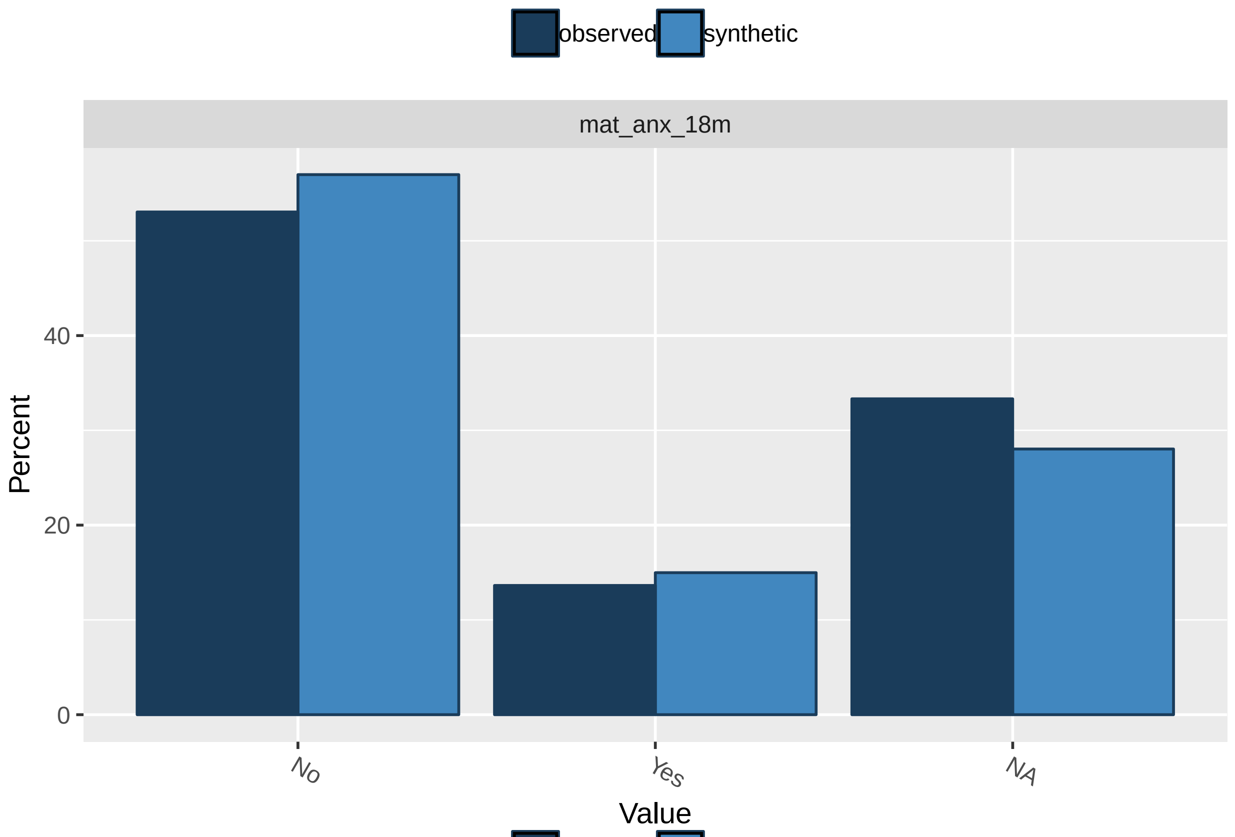
observed
synthetic
pat_pres_10
Percent
0
10
20
30
40
No
Yes
NA
Value

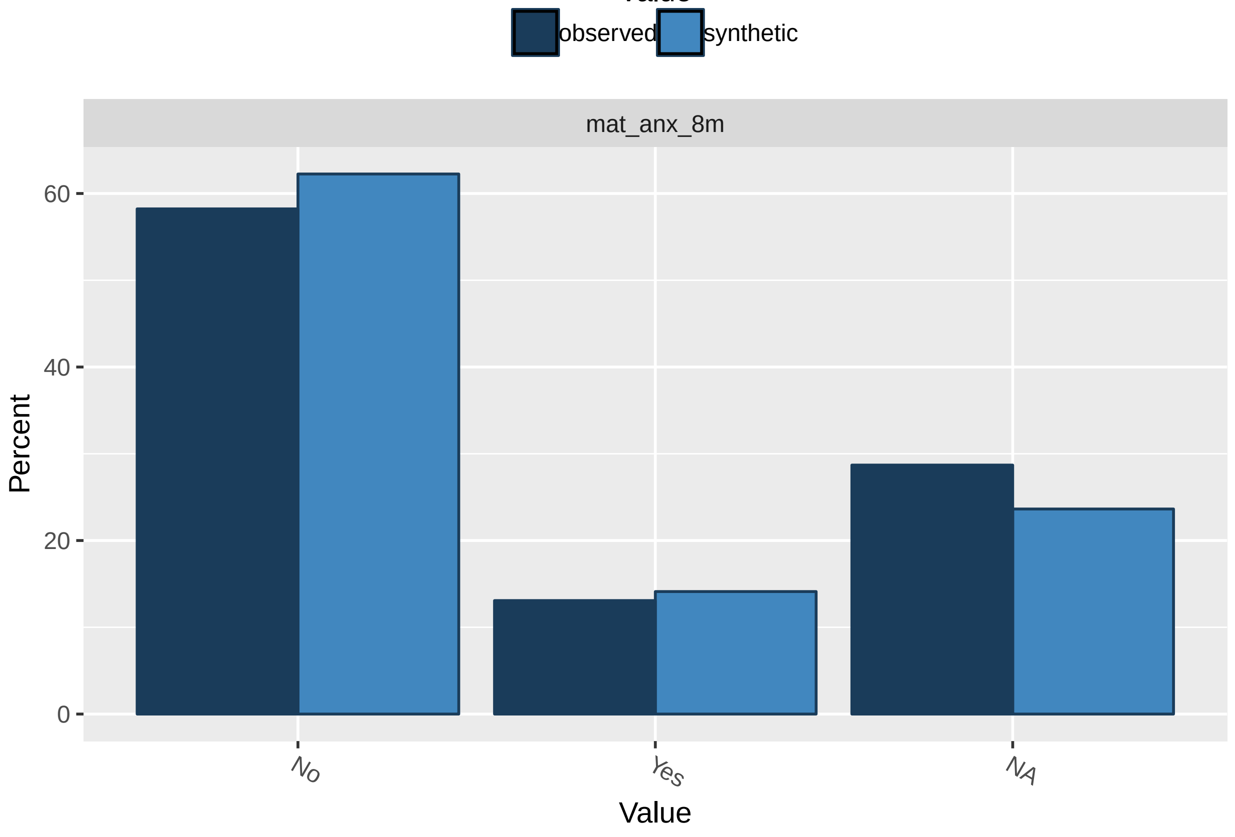

observed
synthetic
pat_pres_8
Percent
0
10
20
30
40
No
Yes
NA
Value

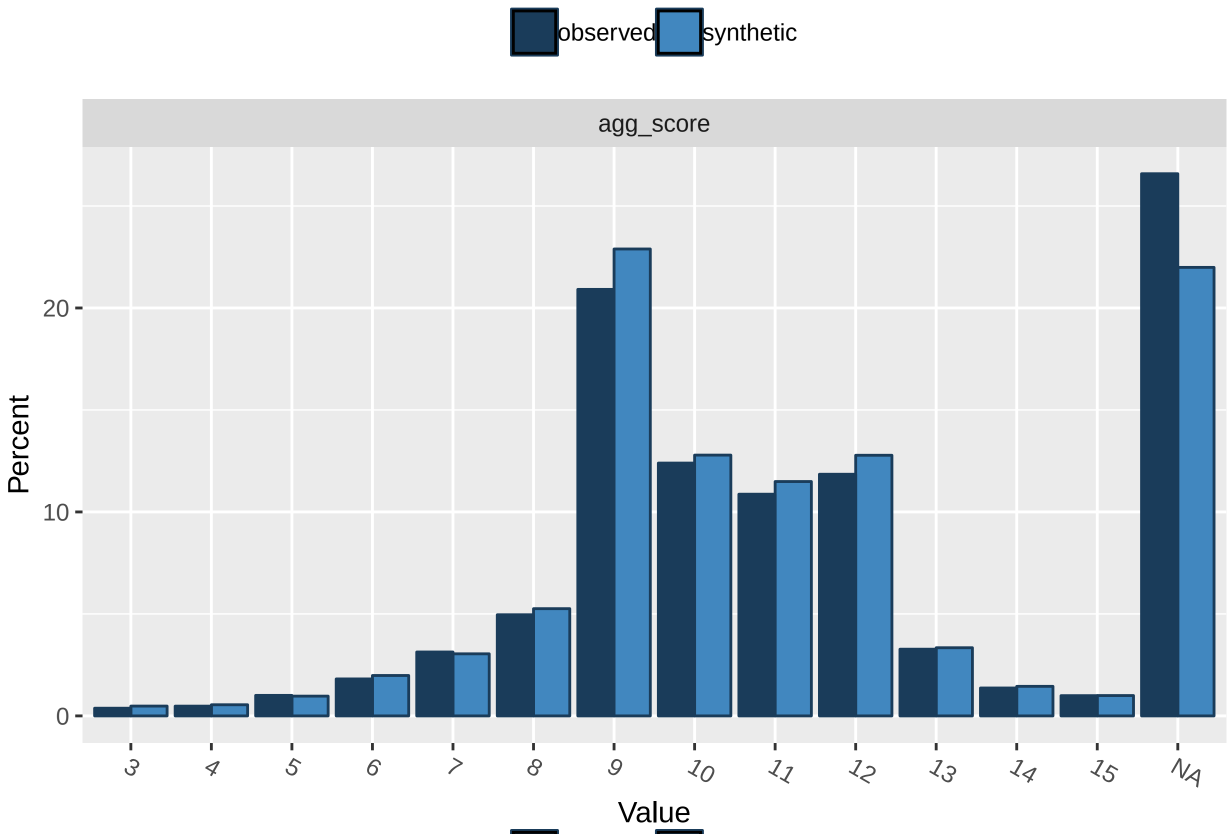

observed
synthetic
pat_pres
Percent
0
20
40
60
No
Yes
NA
Value

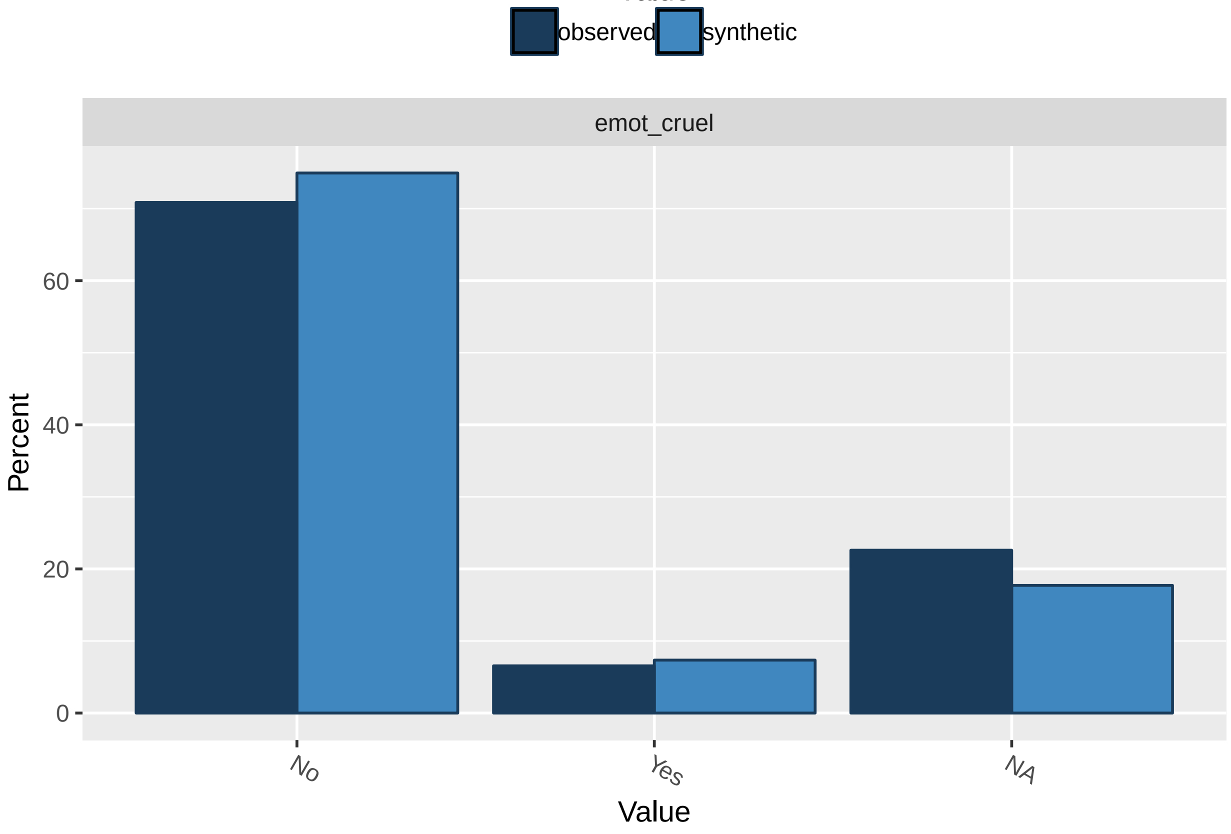

observed
synthetic
num_home
Percent
0
10
20
30
1
2
3
4
5
6
7
8
9 or more
NA
Value

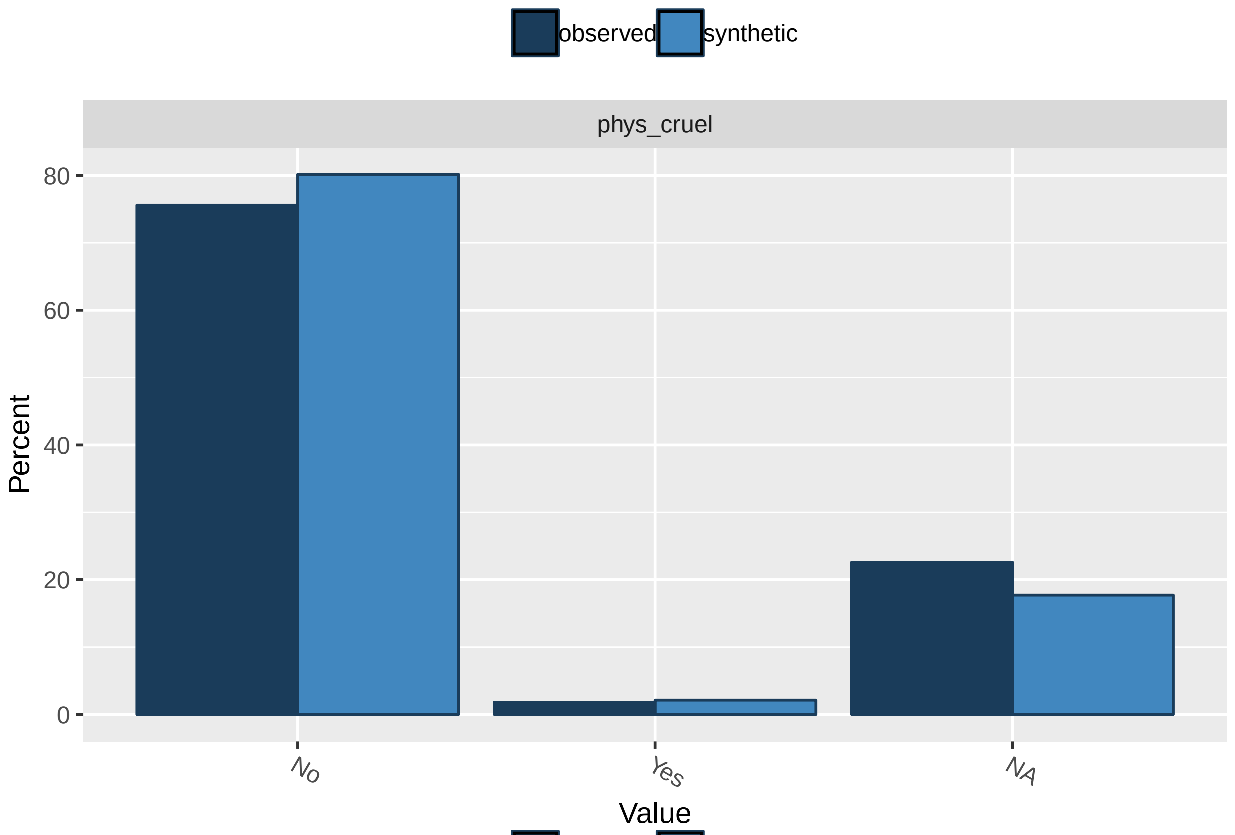

observed
synthetic
mat_anx_1
Percent
0
20
40
No
Yes
NA
Value

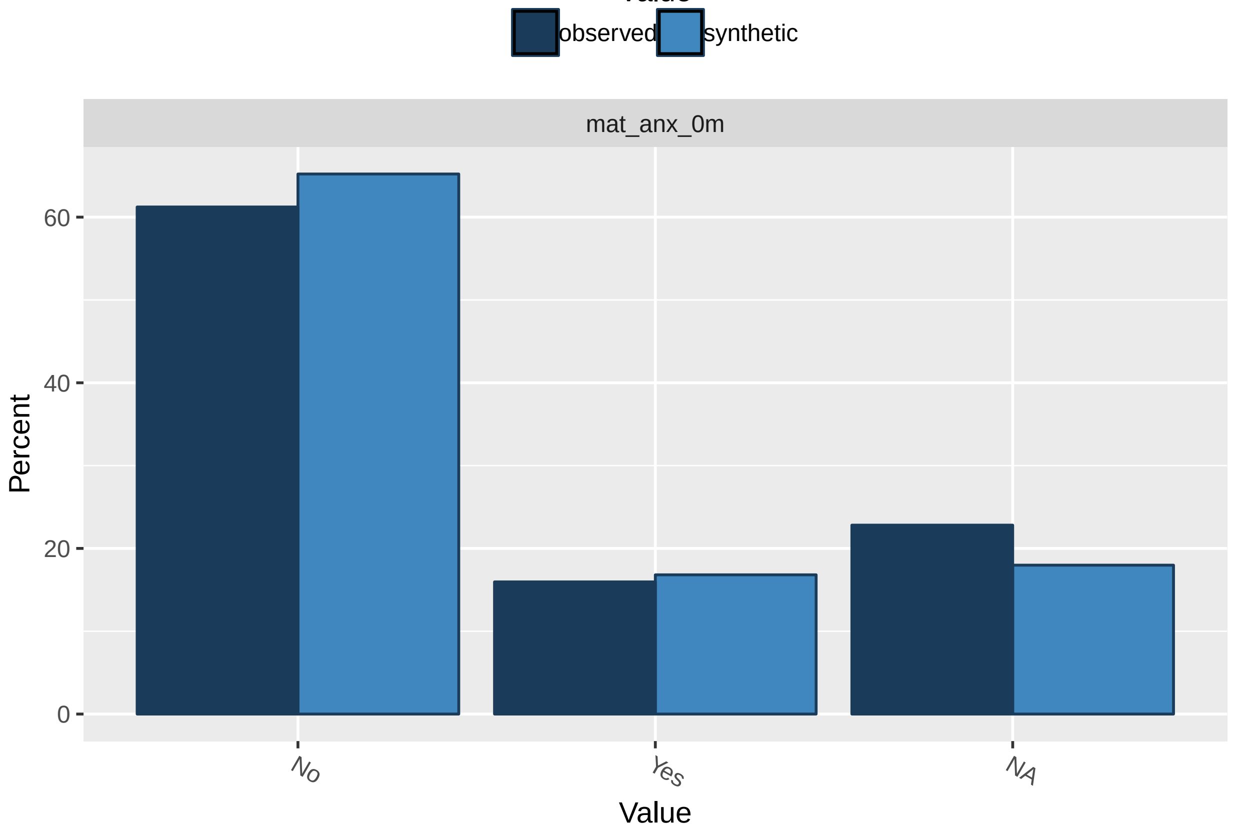

observed
synthetic
mat_anx_18m
Percent
0
20
40
No
Yes
NA
Value

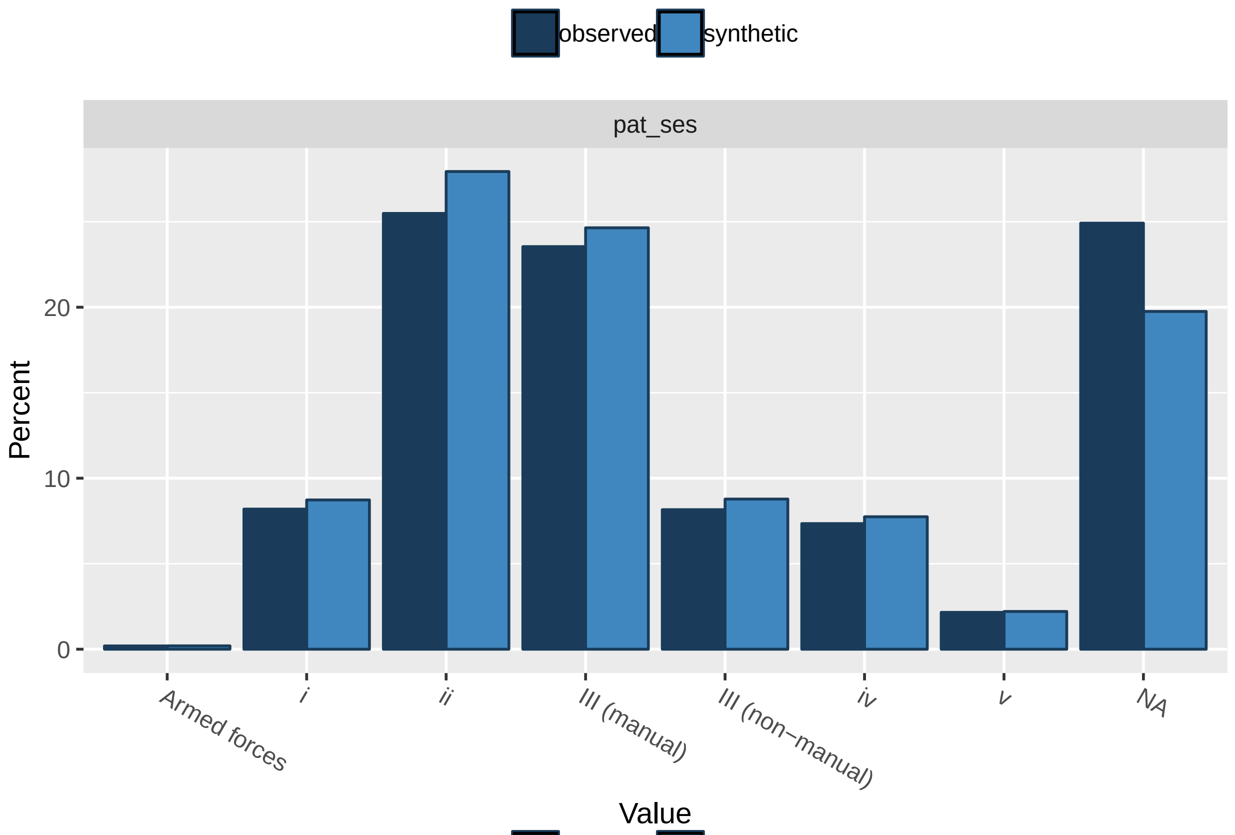

observed
synthetic
mat_anx_8m
Percent
0
20
40
60
No
Yes
NA
Value


observed
synthetic
agg_score
Percent
0
10
20
3
4
5
6
7
8
9
10
11
12
13
14
15
NA
Value

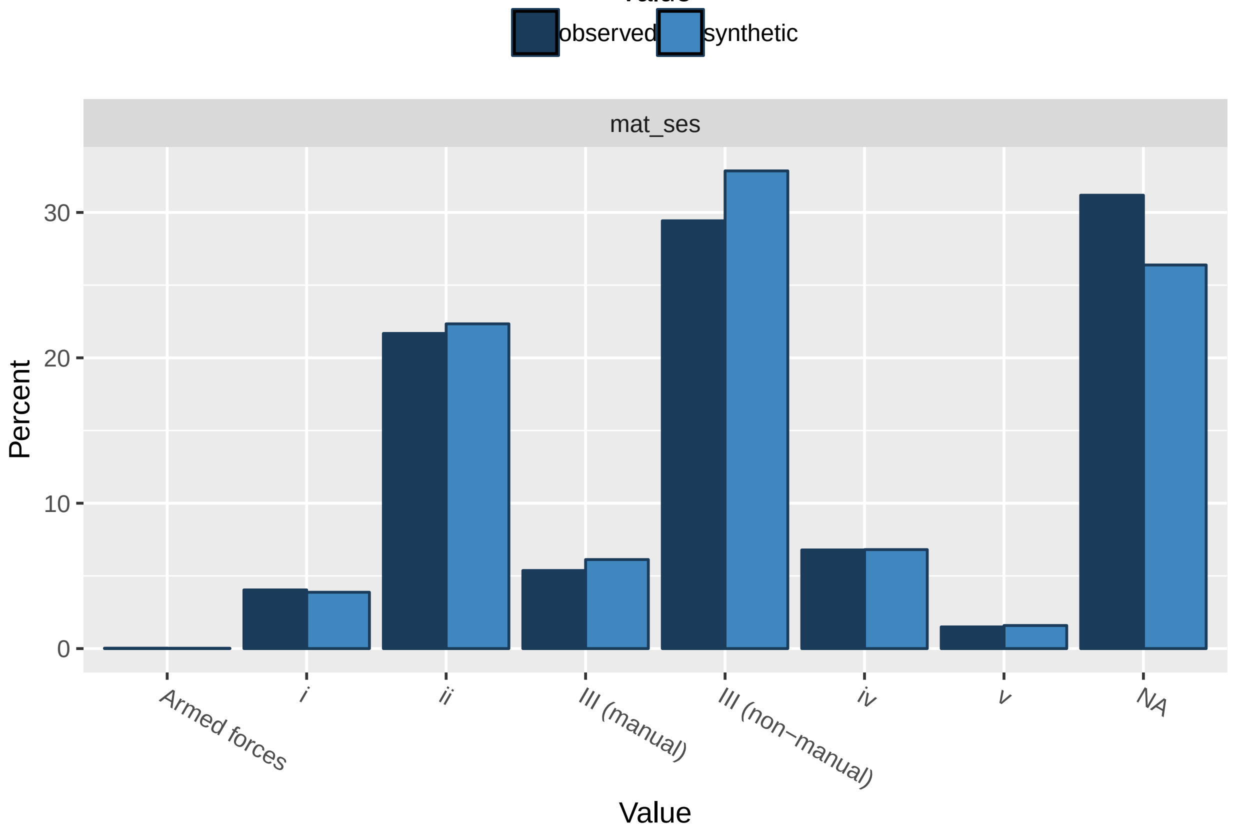

observed
synthetic
emot_cruel
Percent
0
20
40
60
No
Yes
NA
Value


observed
synthetic
phys_cruel
Percent
0
20
40
60
80
No
Yes
NA
Value

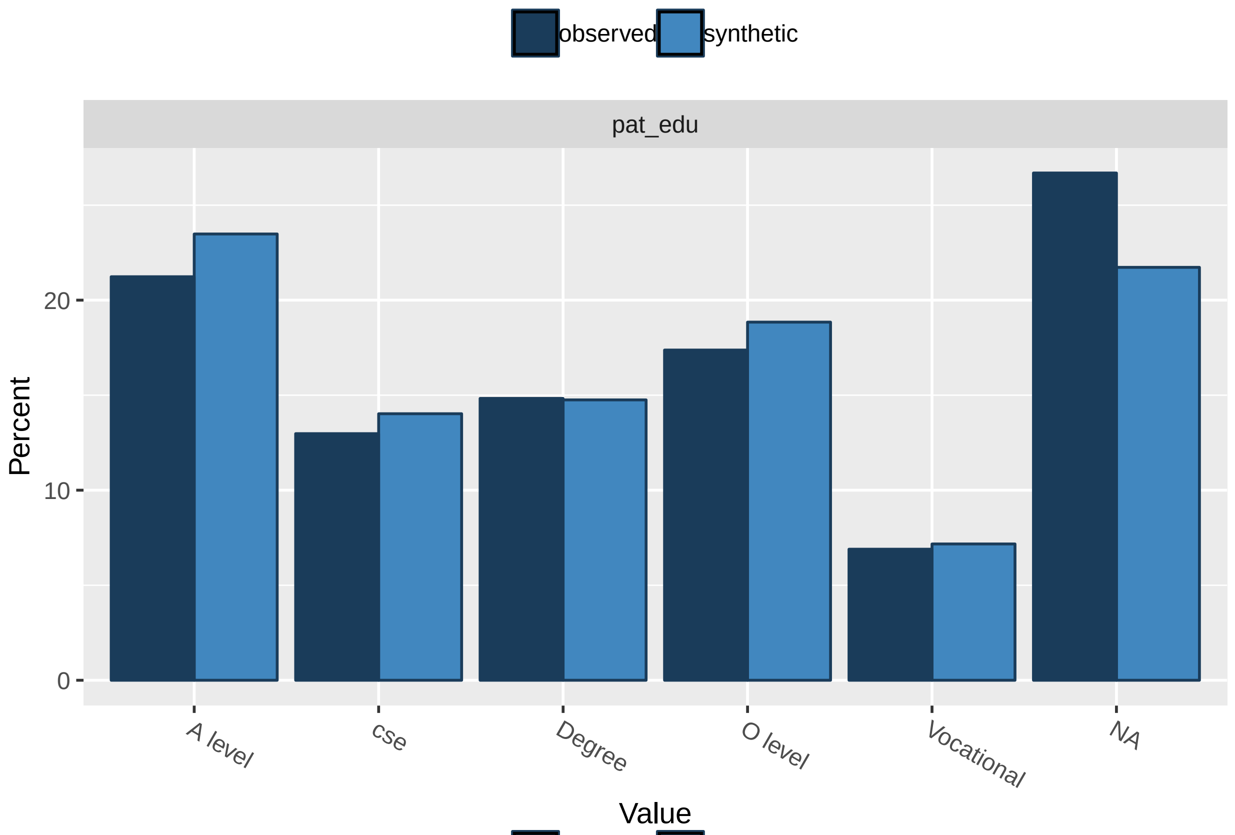

observed
synthetic
mat_anx_0m
Percent
0
20
40
60
No
Yes
NA
Value

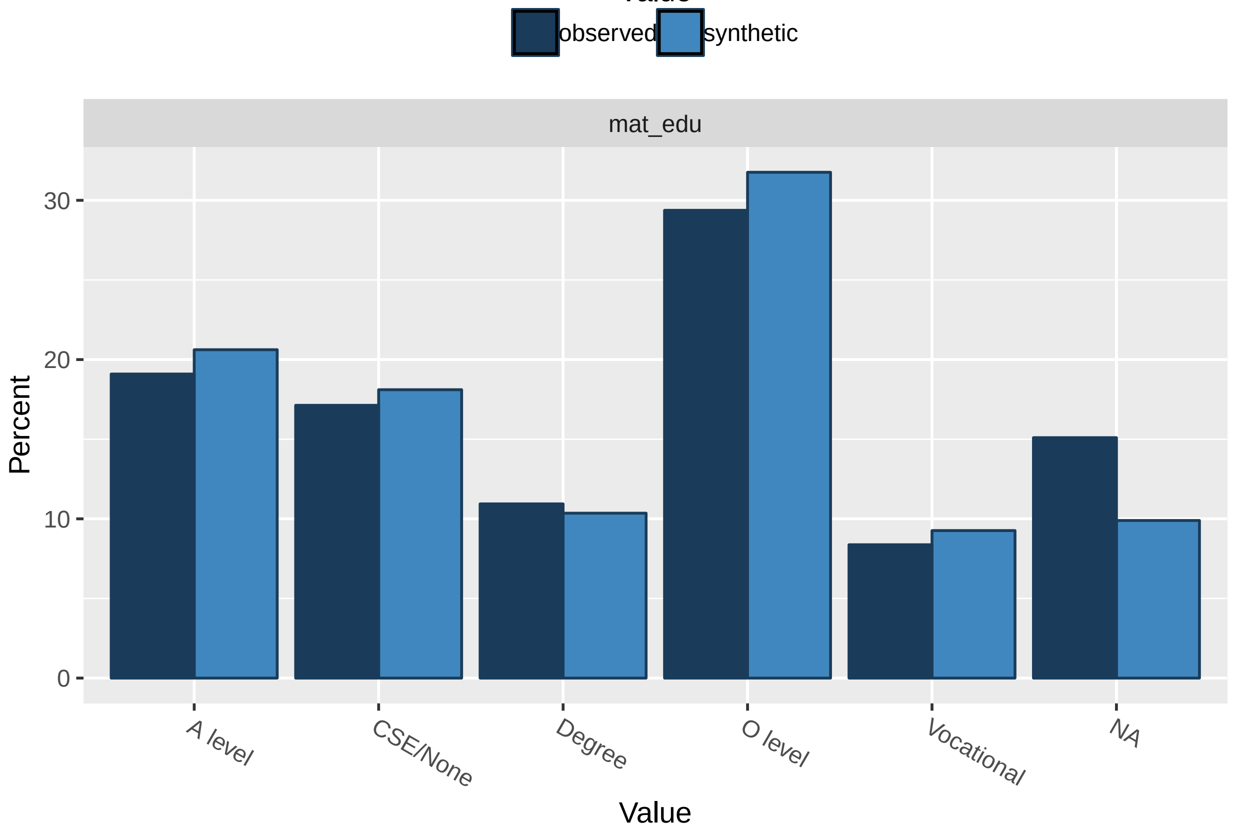


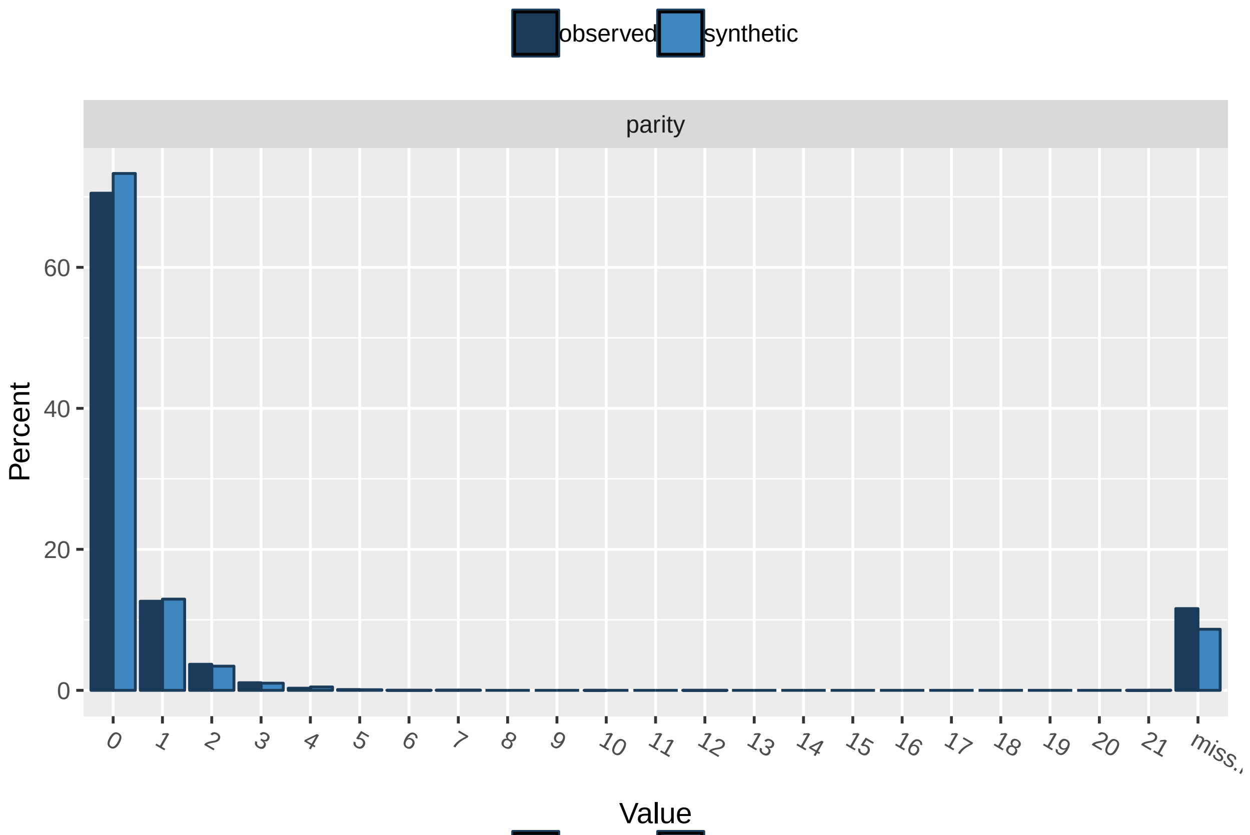
observed
synthetic
pat_ses
Percent
0
10
20
Armed forces
i
ii
III (manual)
III (non−manual)
iv
v
NA
Value

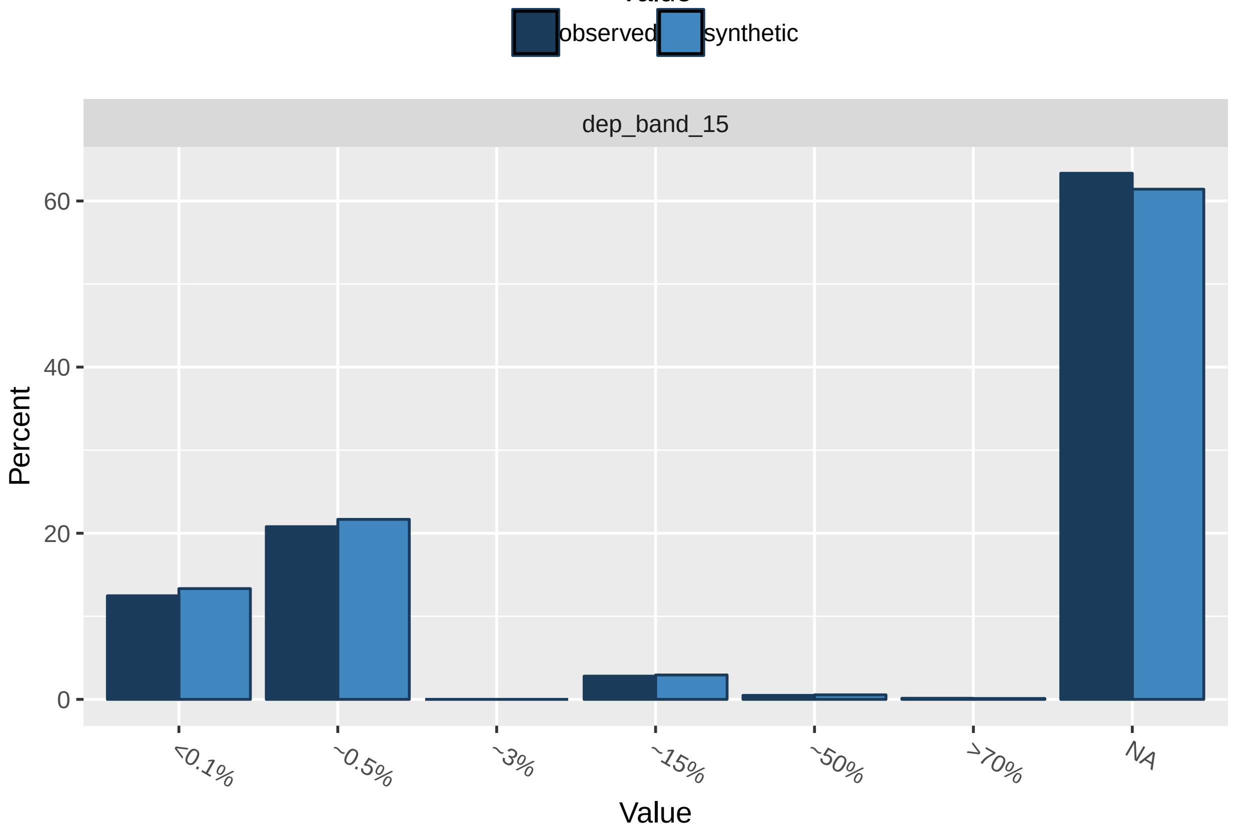


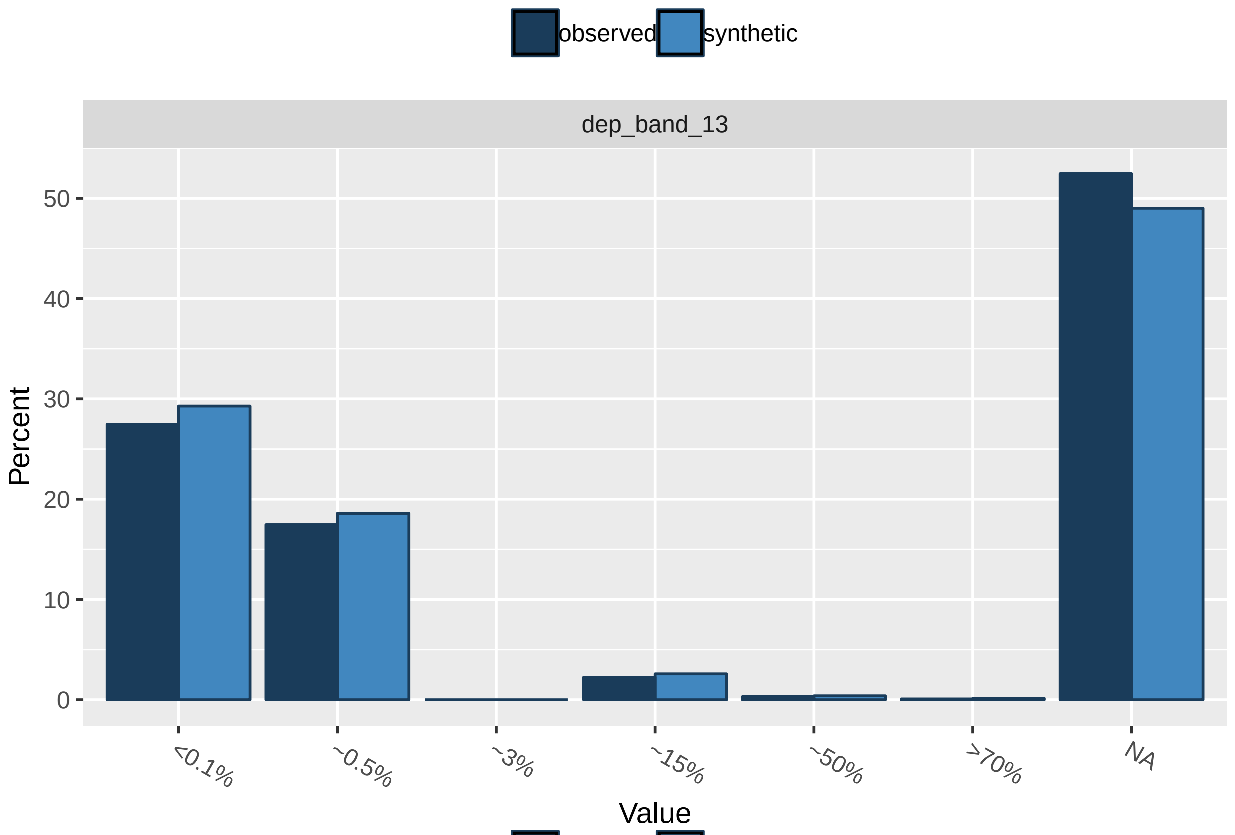
observed
synthetic
mat_ses
Percent
0
10
20
30
Armed forces
i
ii
III (manual)
III (non−manual)
iv
v
NA
Value

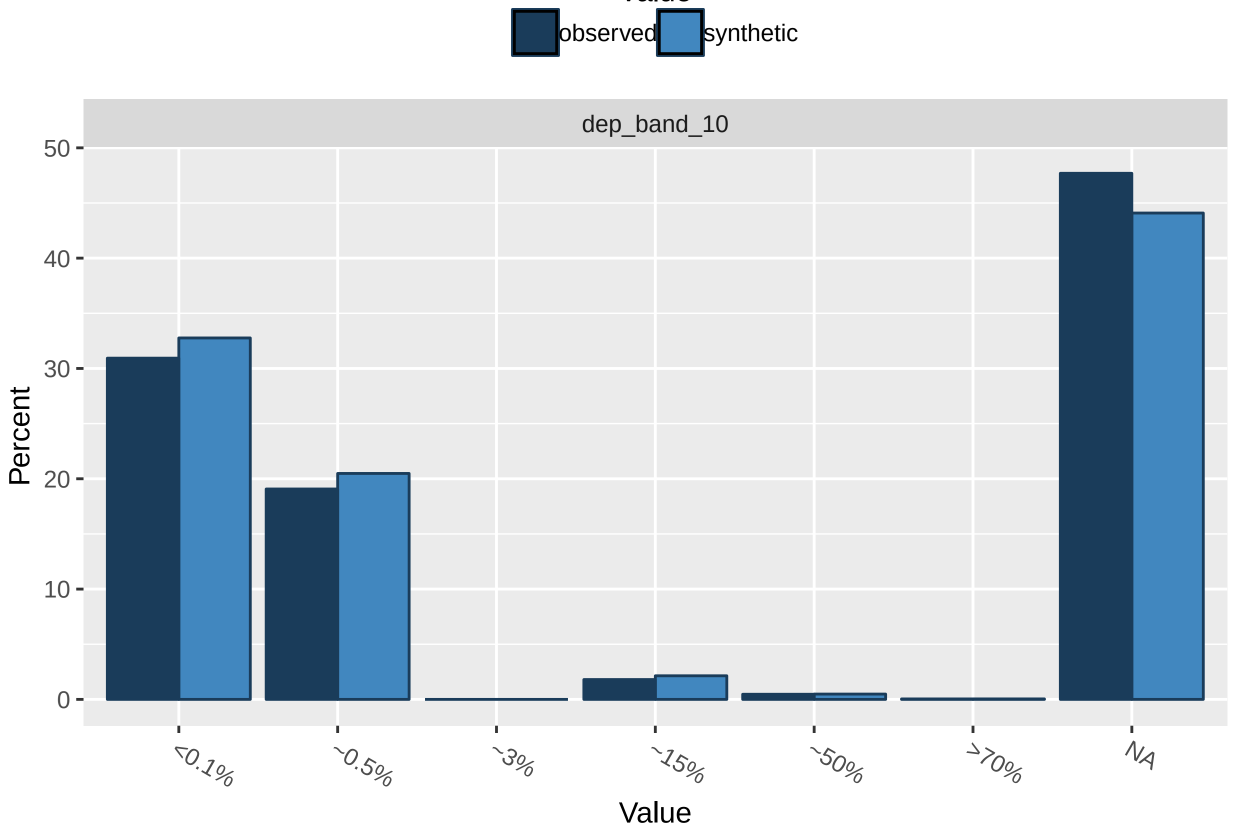

observed
synthetic
pat_edu
Percent
0
10
20
A level
cse
Degree
O level
Vocational
NA
Value

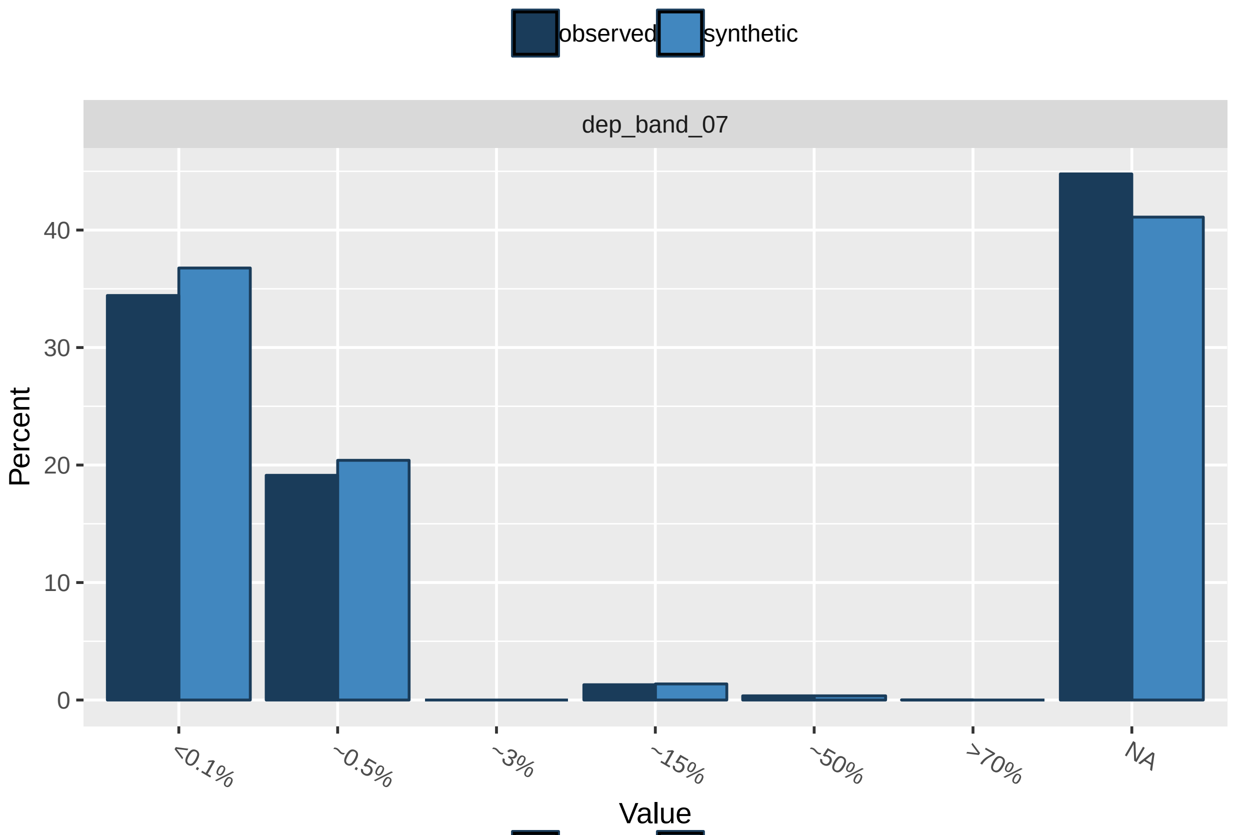

observed
synthetic
mat_edu
Percent
0
10
20
30
A level
CSE/None
Degree
O level
Vocational
NA
Value

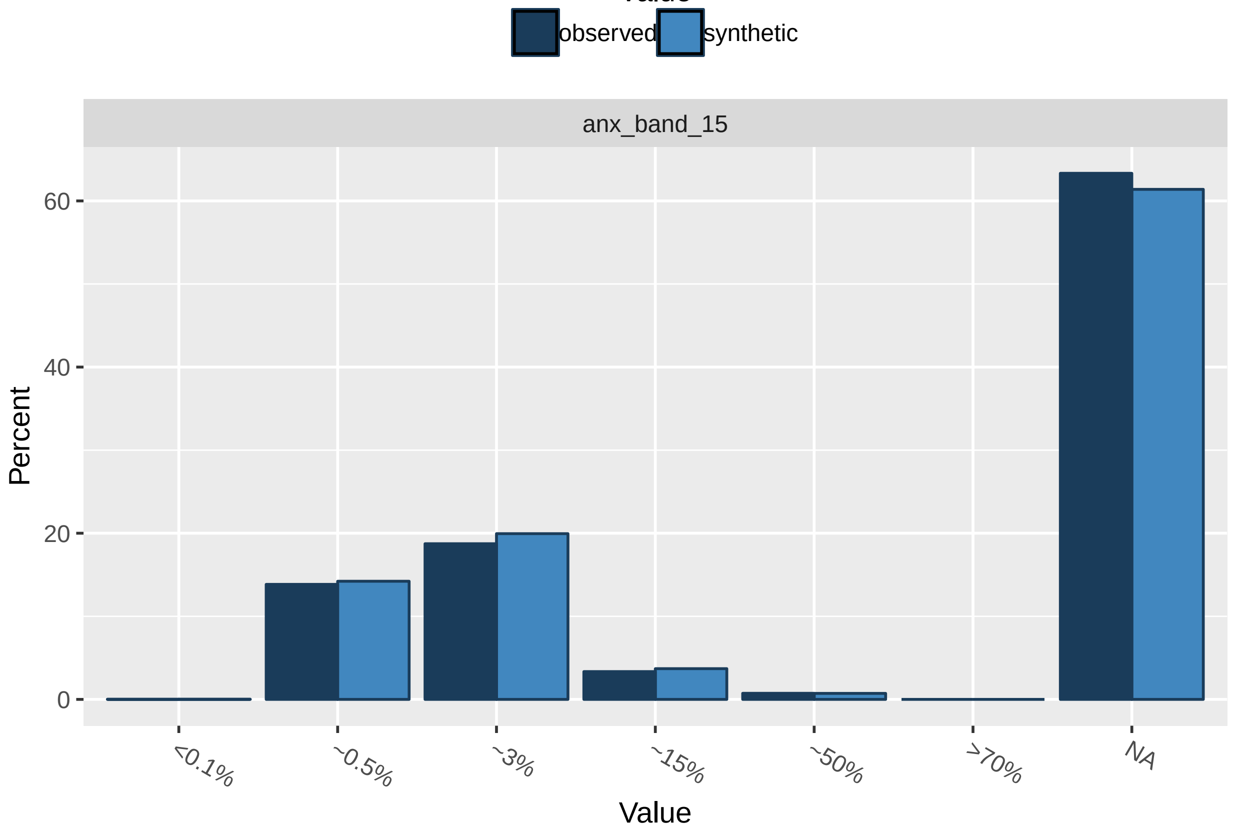

observed
synthetic
parity
Percent
60
40
20
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
miss.
Value

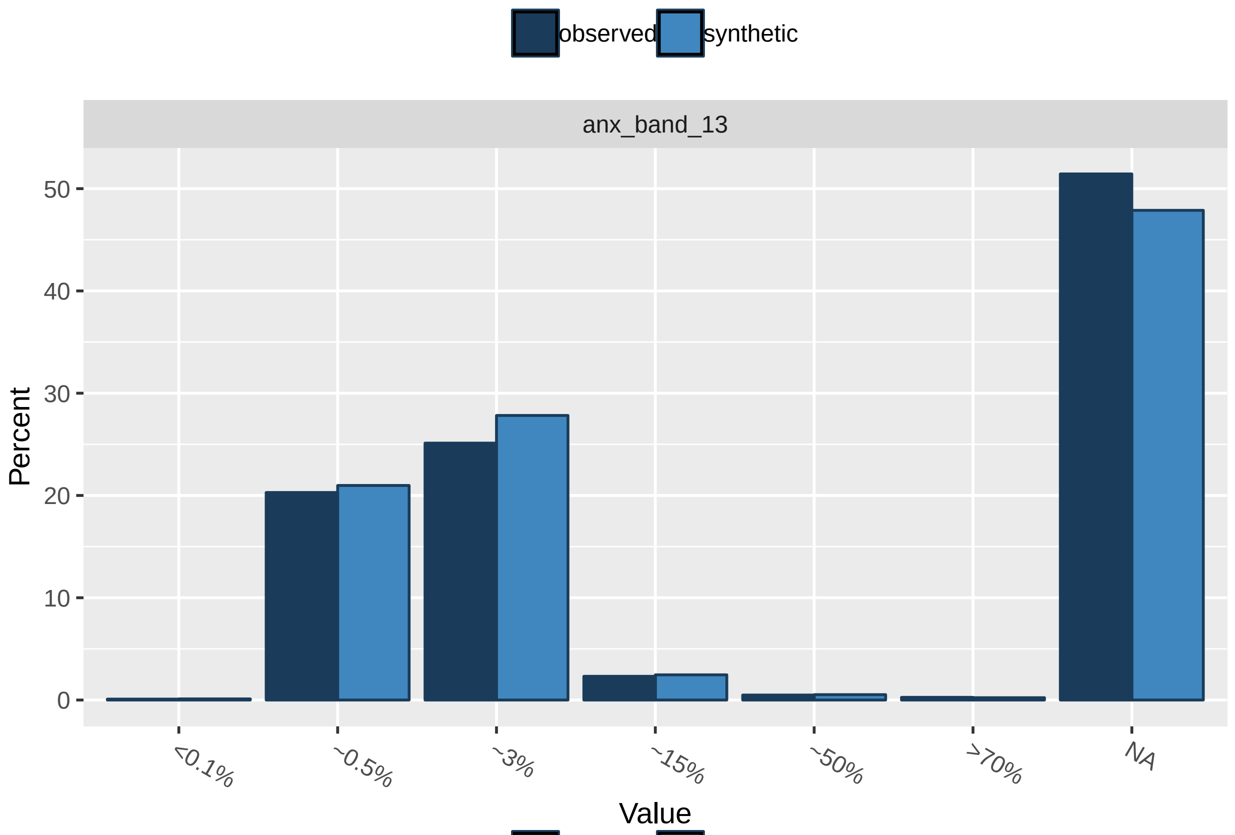

observed
synthetic
dep_band_15
Percent
60
40
20
0
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

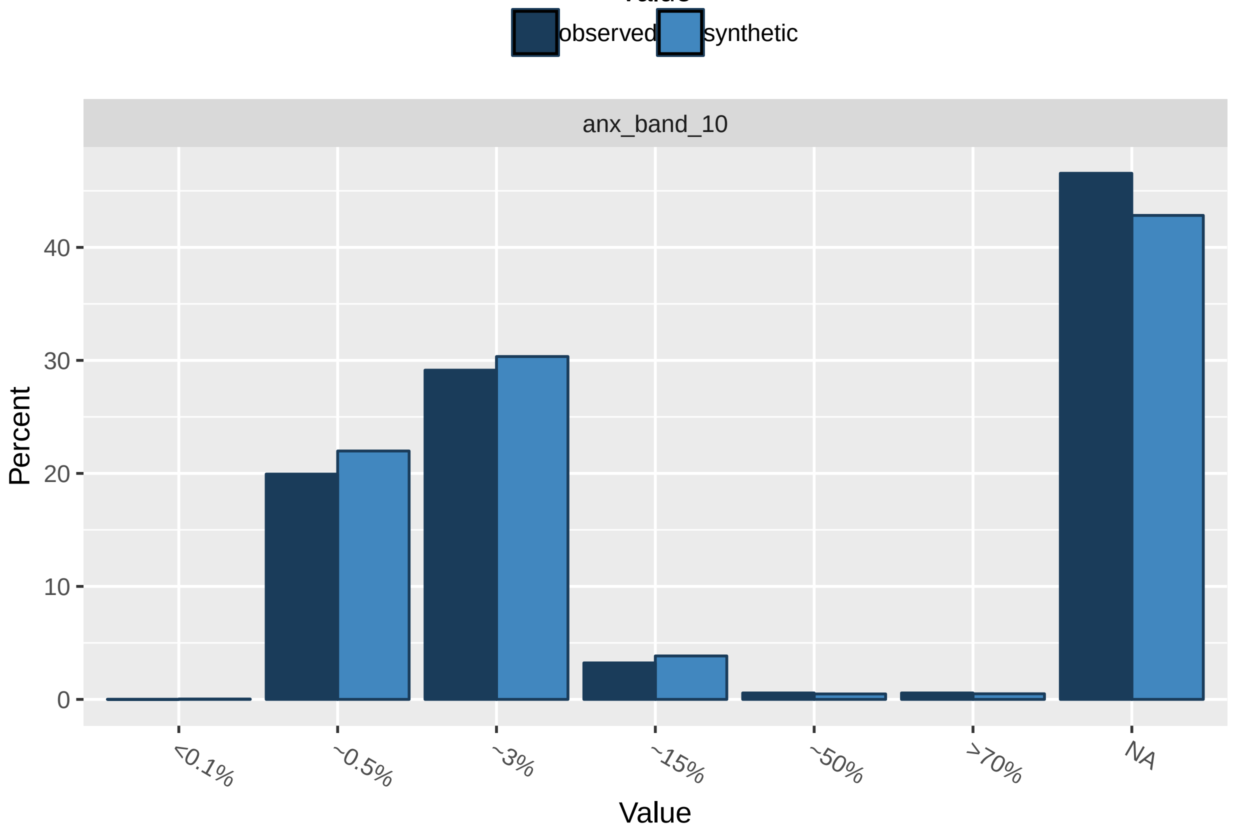

observed
synthetic
dep_band_13
Percent
50
40
30
20
10
0
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

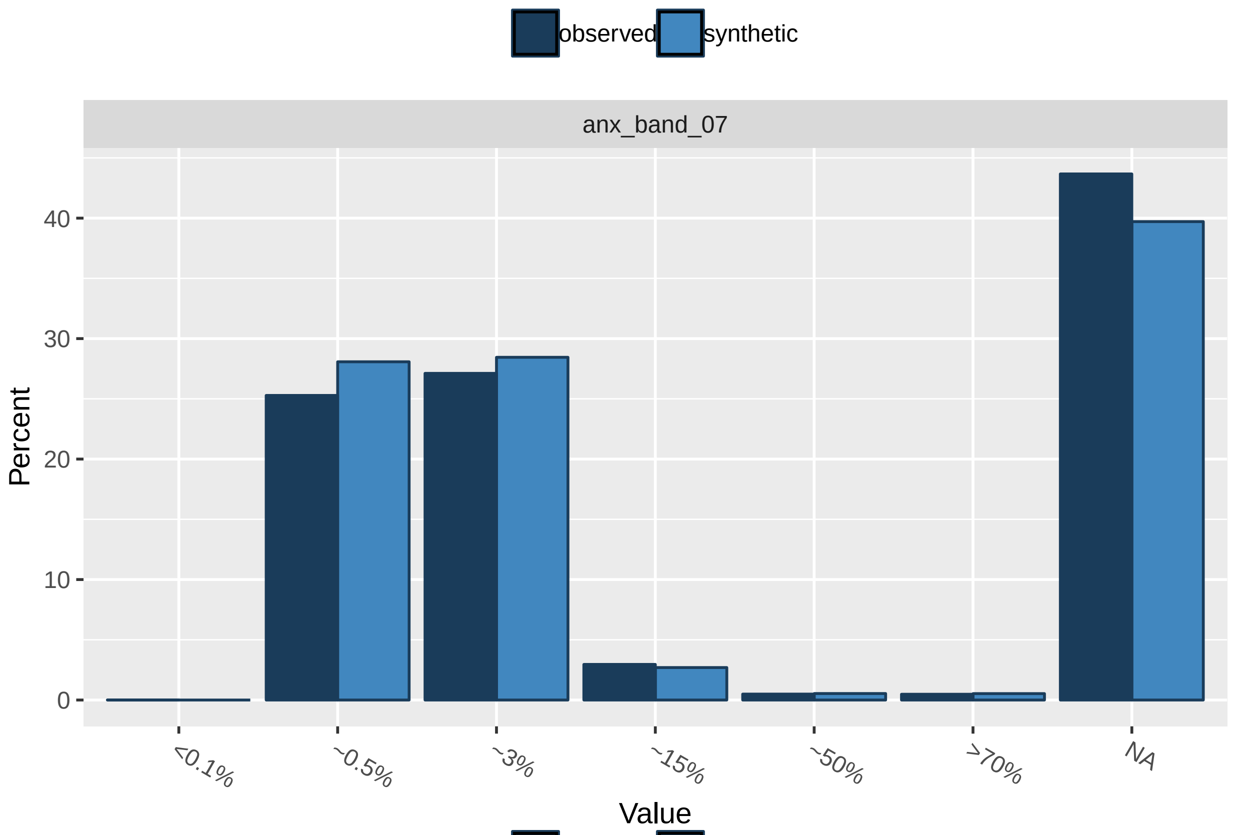

observed
synthetic
dep_band_10
Percent
50
40
30
20
10
0
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

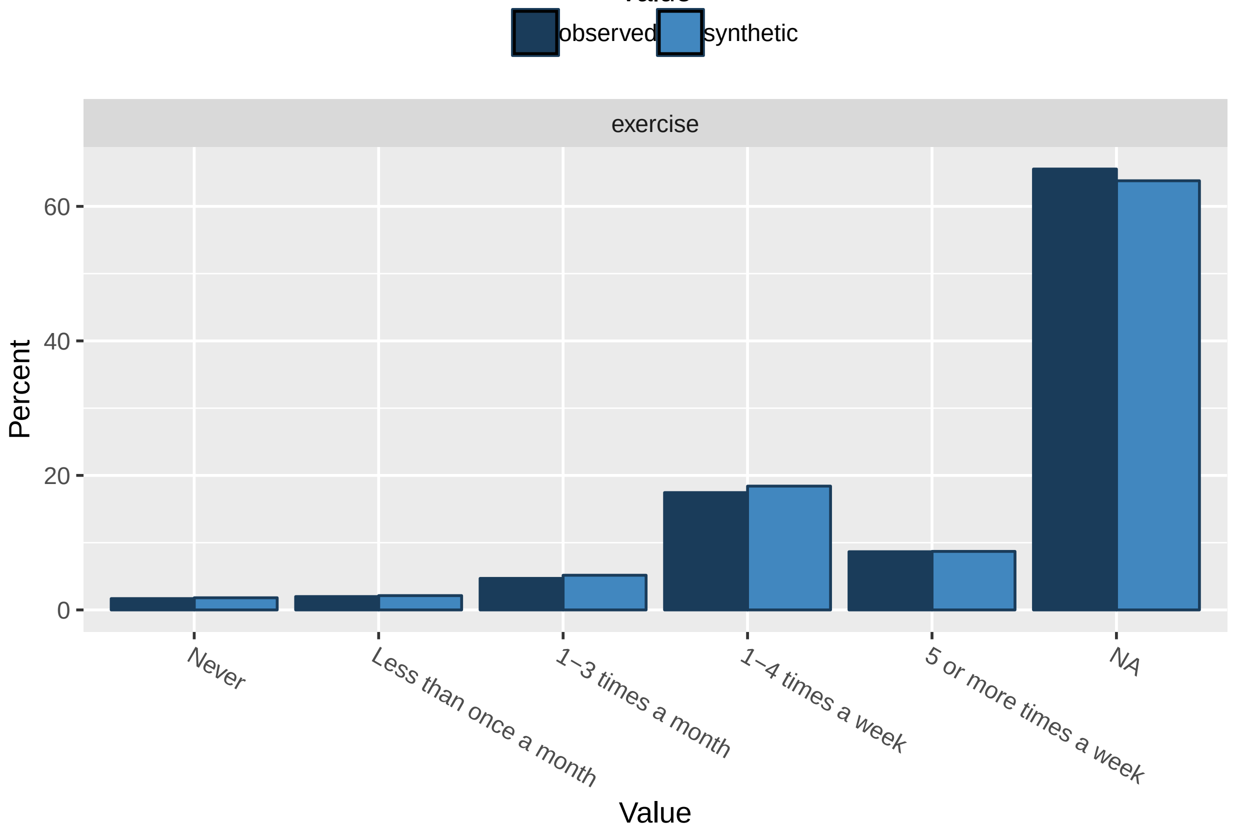

observed
synthetic
dep_band_07
Percent
0
10
20
30
40
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

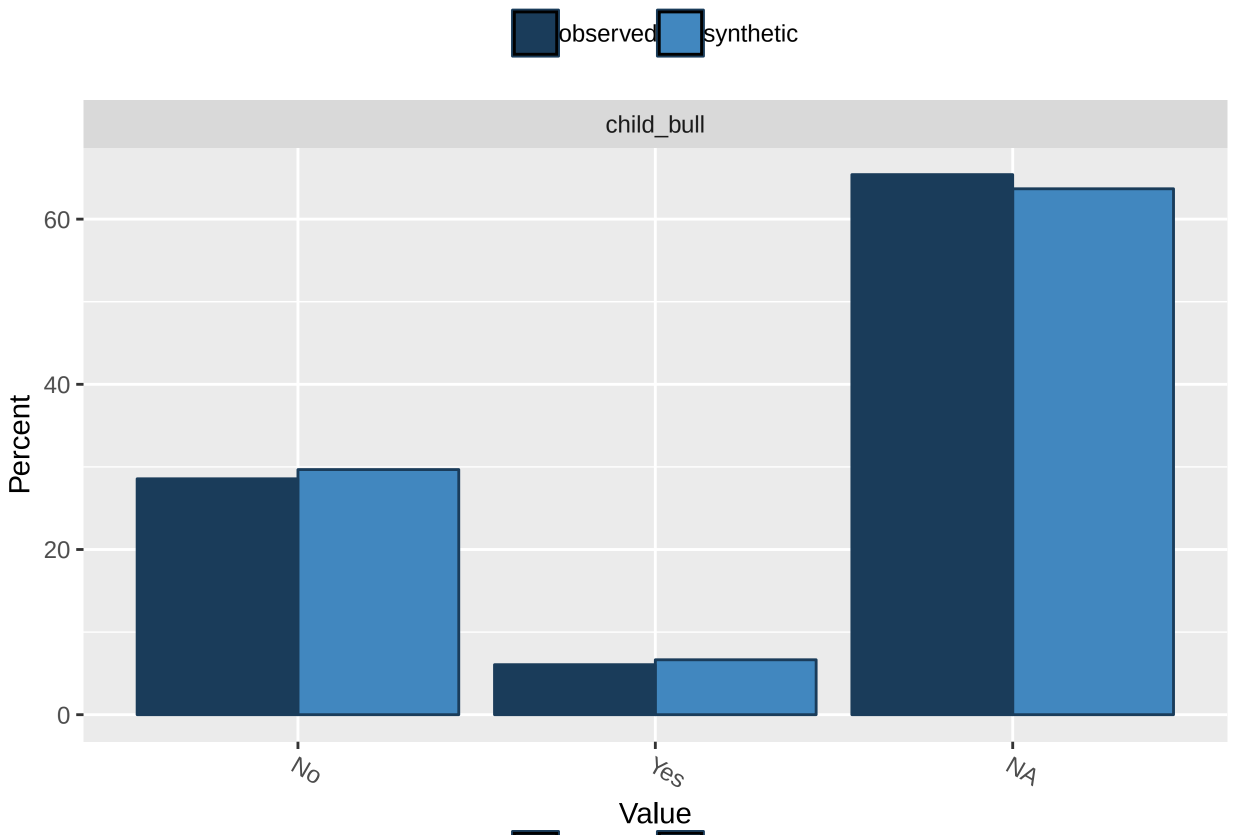

observed
synthetic
anx_band_15
Percent
0
20
40
60
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

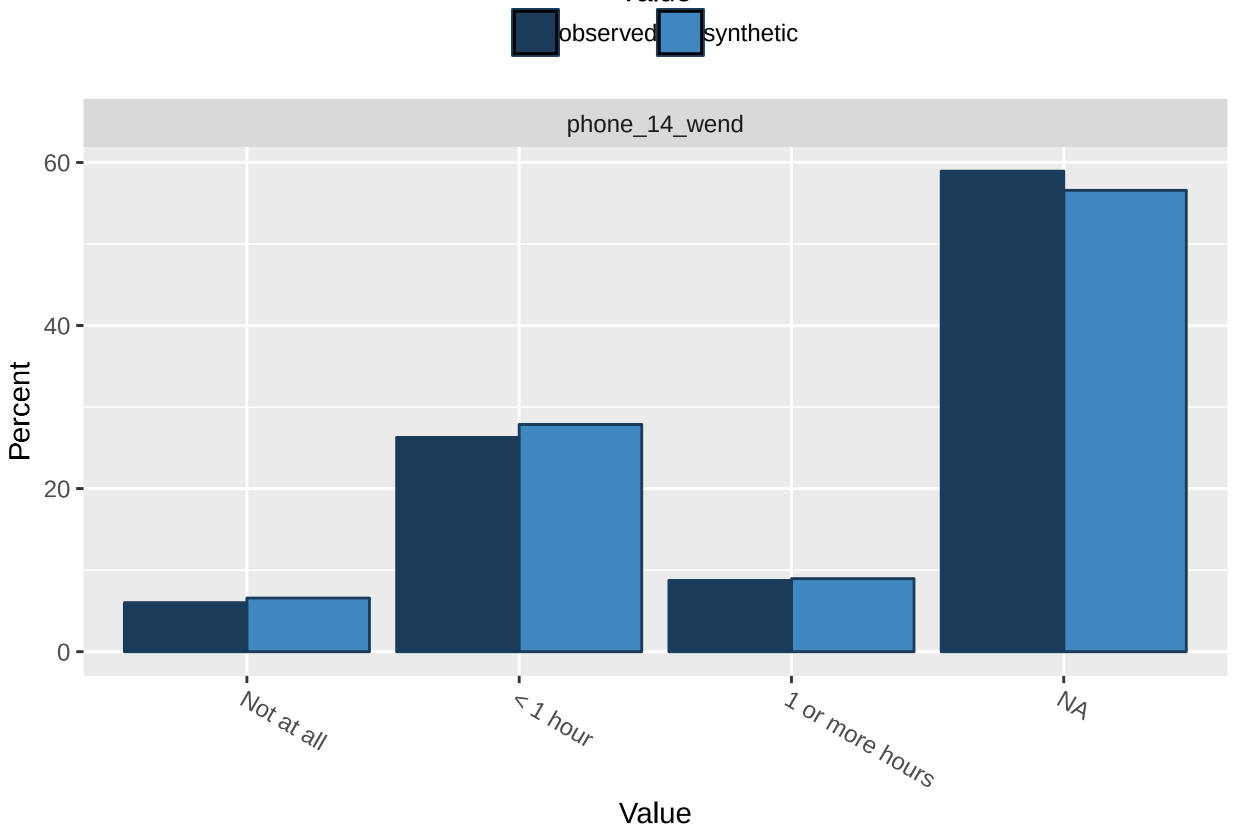

observed
synthetic
anx_band_13
Percent
50
40
30
20
10
0
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

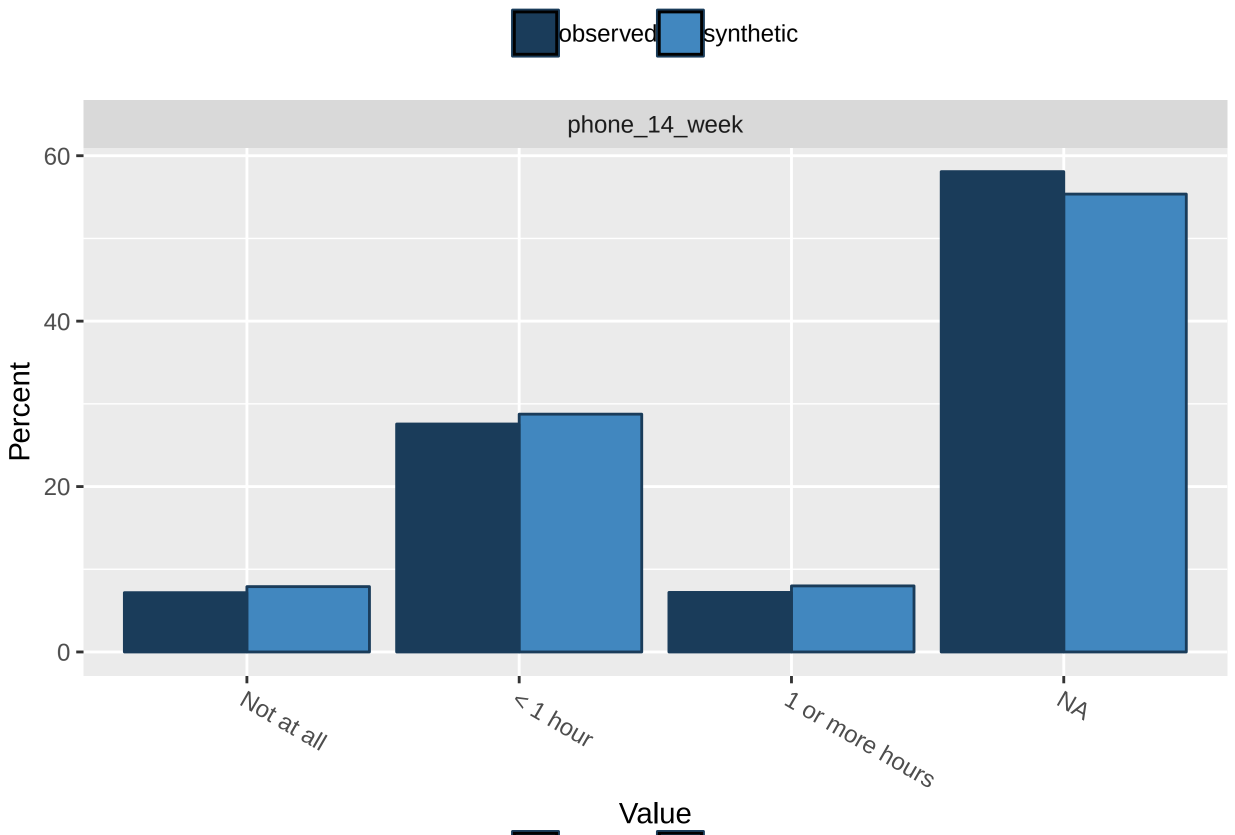

observed
synthetic
anx_band_10
Percent
40
30
20
10
0
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

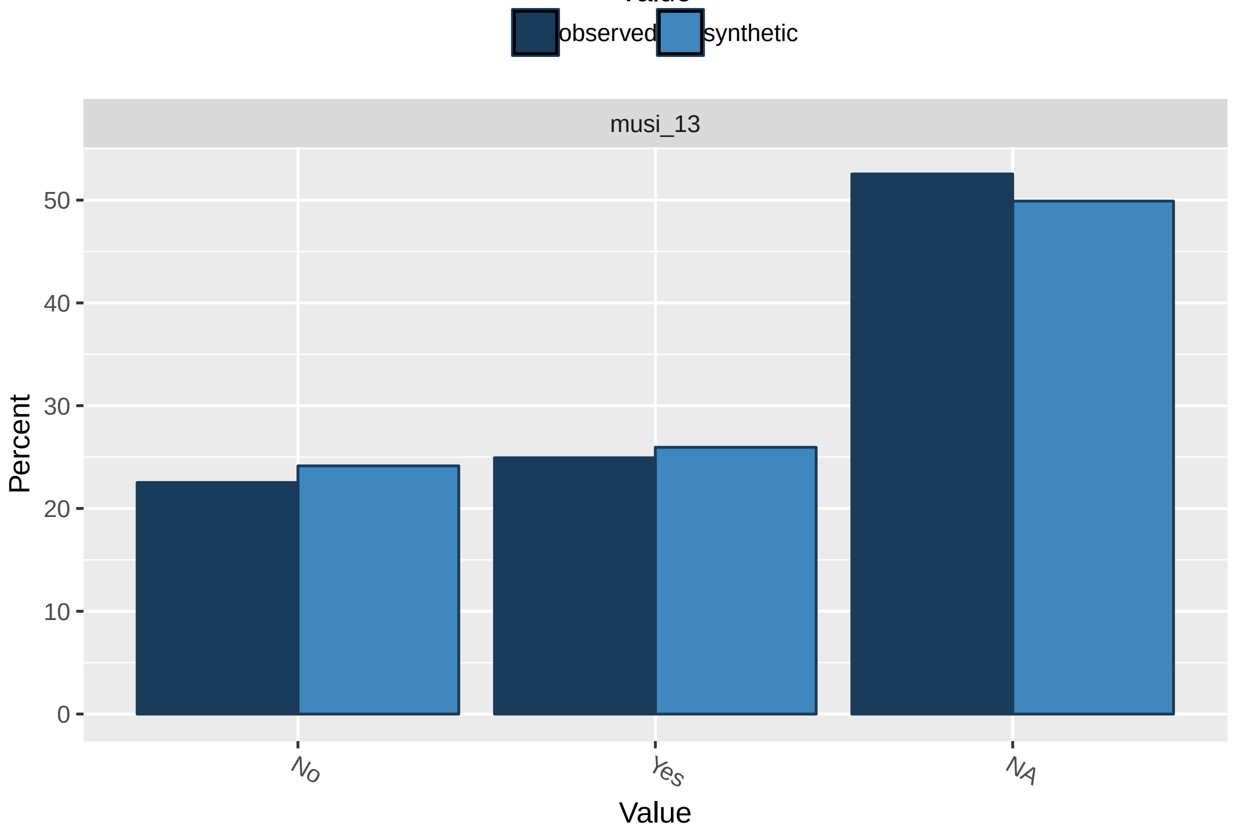

observed
synthetic
anx_band_07
Percent
0
10
20
30
40
<0.1%
~0.5%
~3%
~15%
~50%
>70%
NA
Value

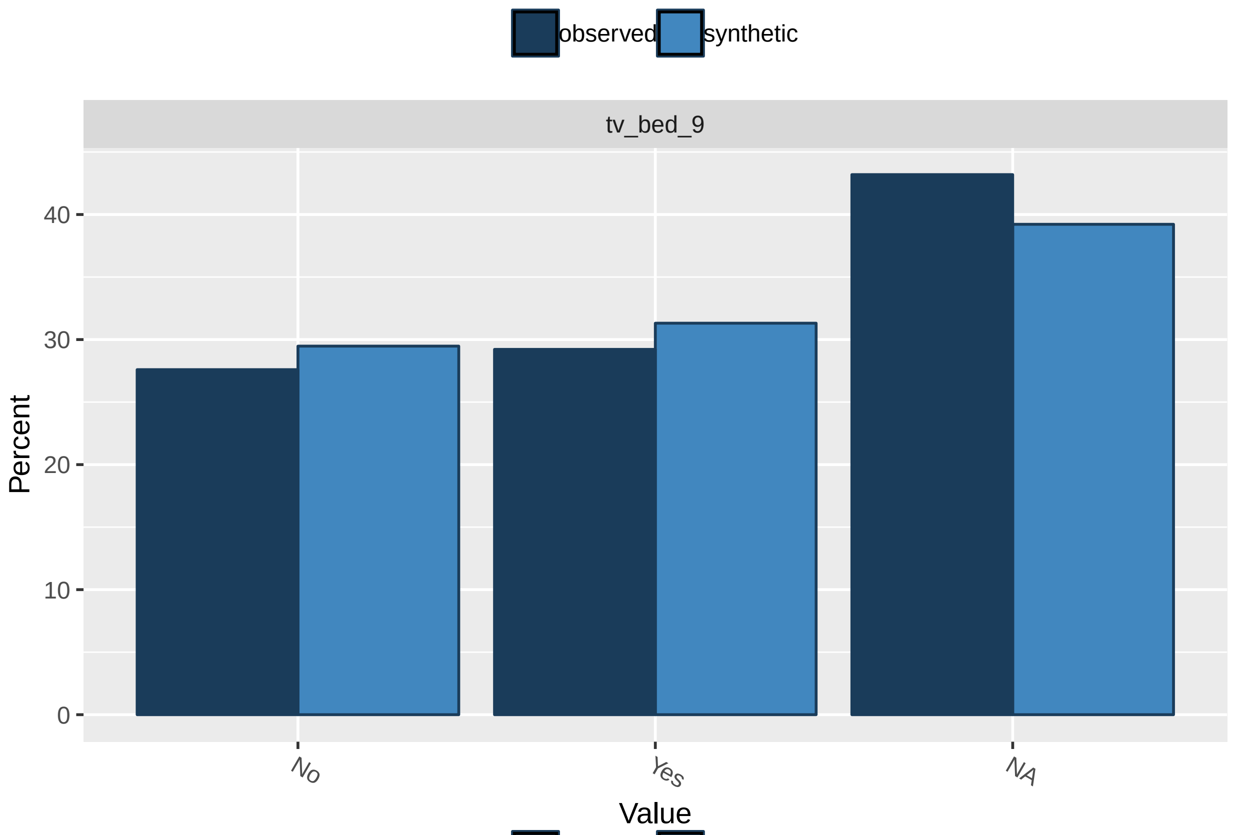

observed
synthetic
exercise
Percent
0
20
40
60
Never
Less than once a month
1–3 times a month
1–4 times a week
5 or more times a week
NA
Value

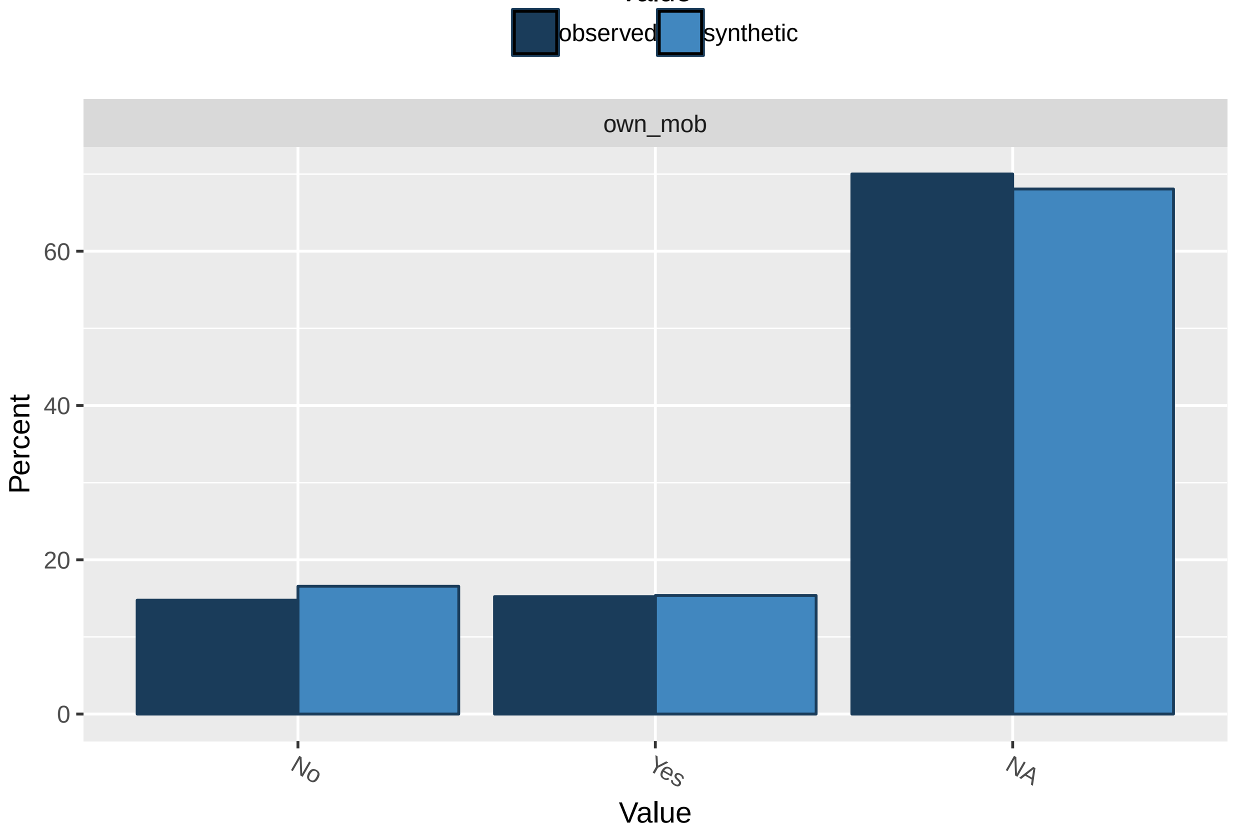

observed
synthetic
child_bull
Percent
60
40
20
0
No
Yes
NA
Value

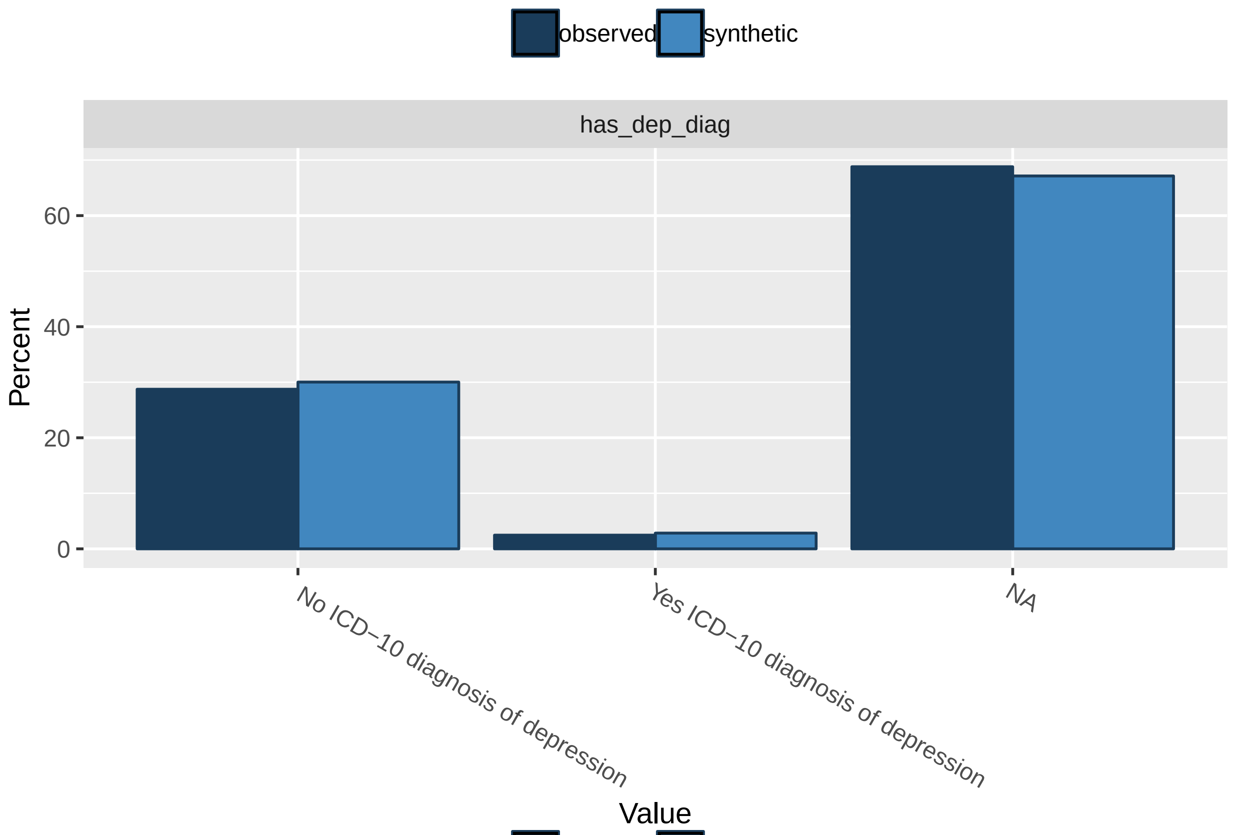

observed
synthetic
phone_14_wend
Percent
60
40
20
0
Not at all
< 1 hour
1 or more hours
NA
Value

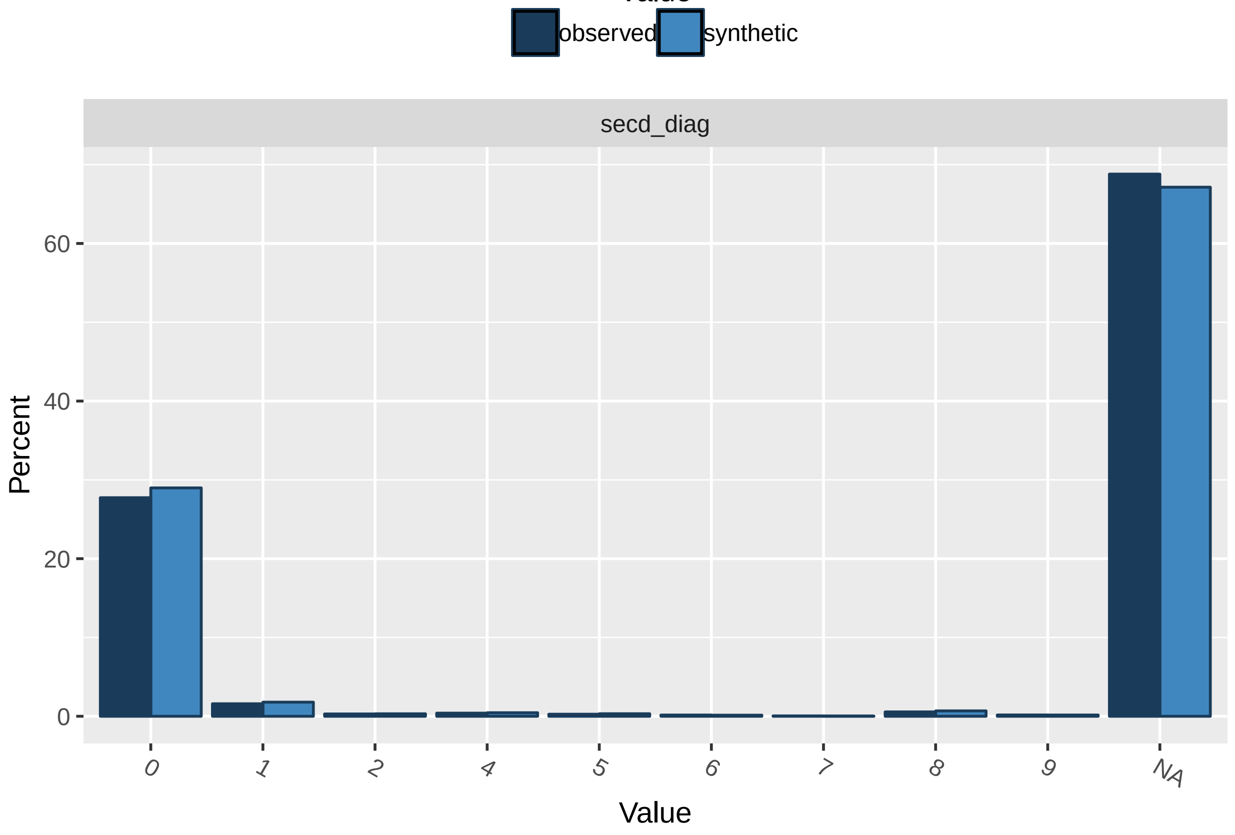

observed
synthetic
phone_14_week
Percent
60
40
20
0
Not at all
< 1 hour
1 or more hours
NA
Value

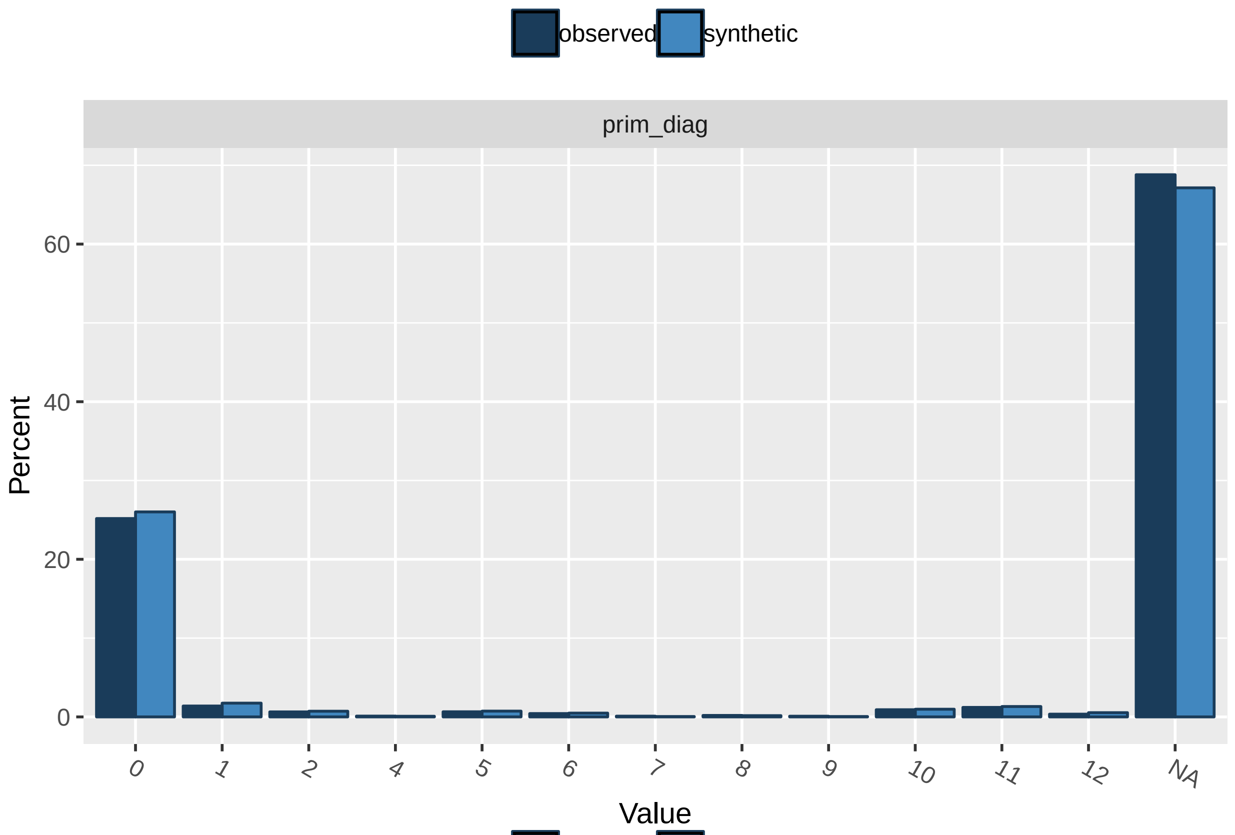

observed
synthetic
musi_13
Percent
50
40
30
20
10
0
No
Yes
NA
Value

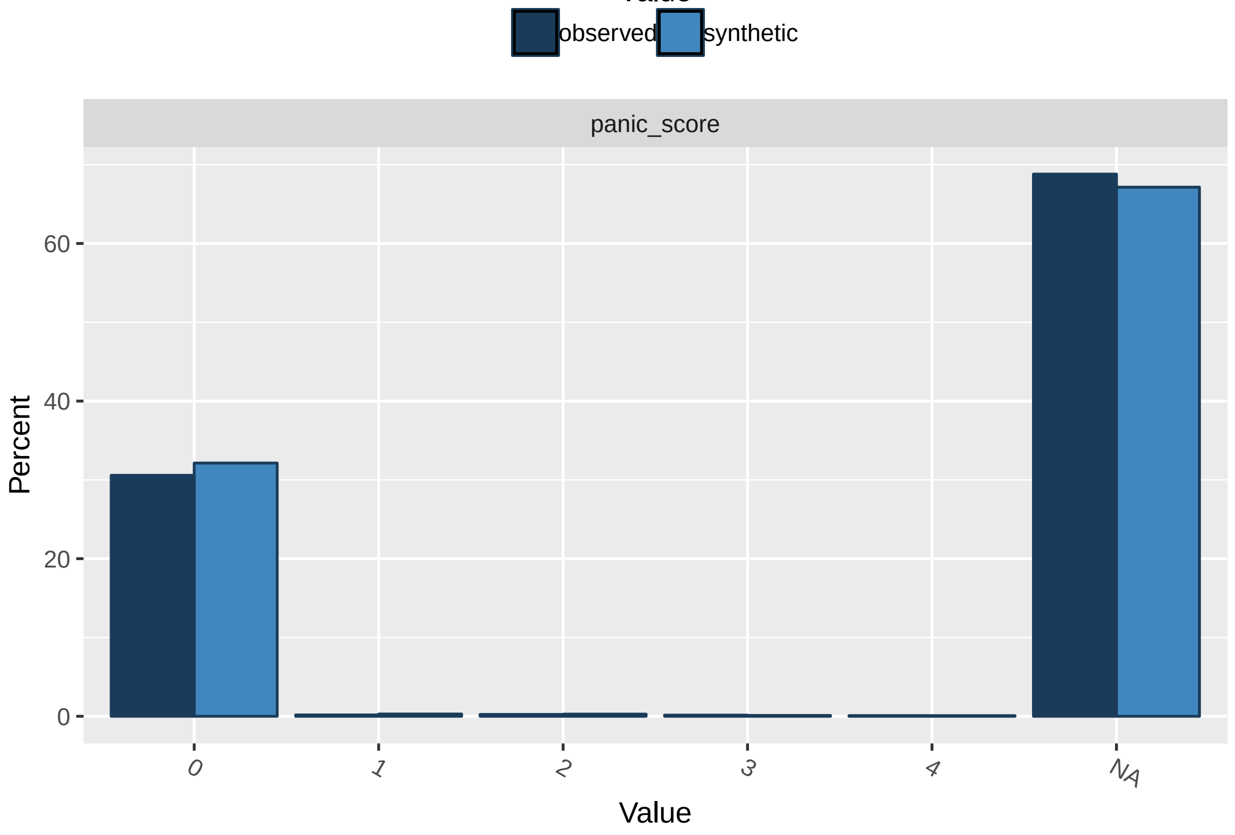

observed
synthetic
tv_bed_9
Percent
0
10
20
30
40
No
Yes
NA
Value

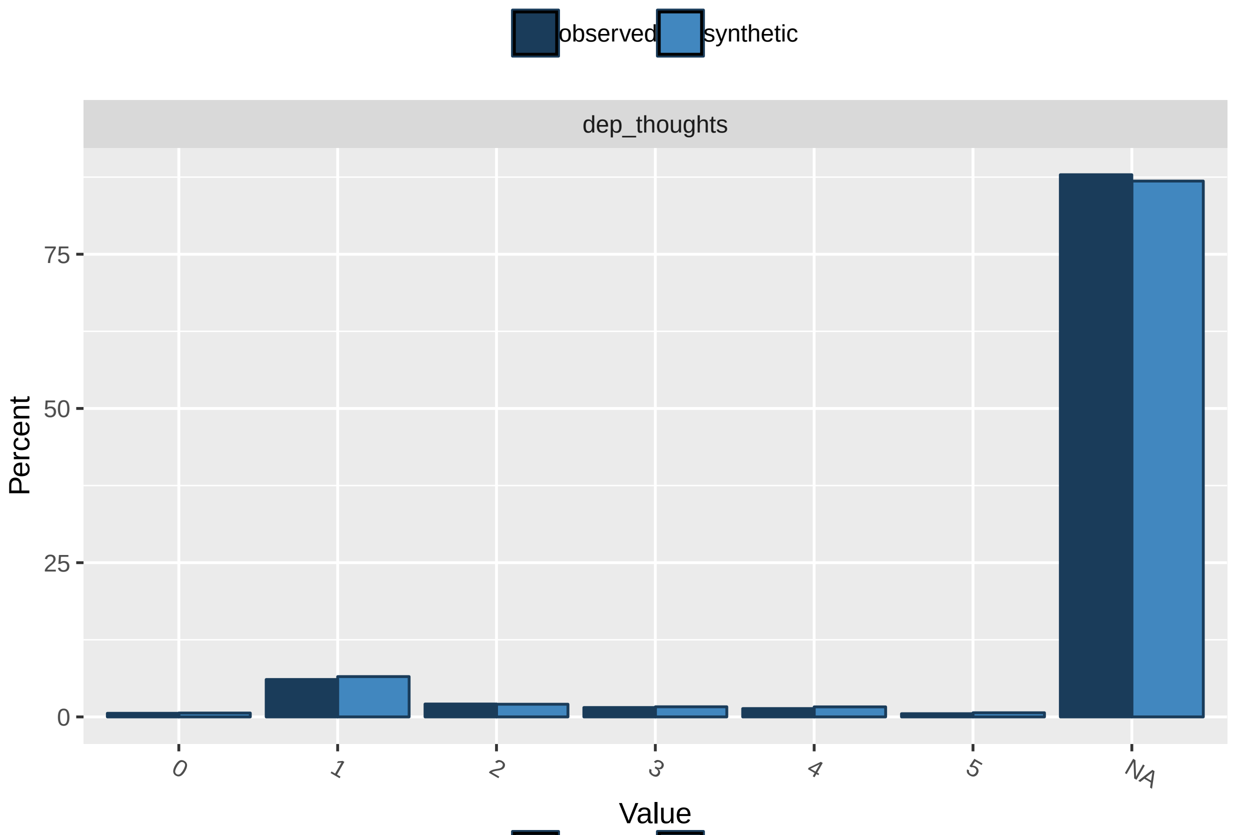

observed
synthetic
own_mob
Percent
0
20
40
60
No
Yes
NA
Value

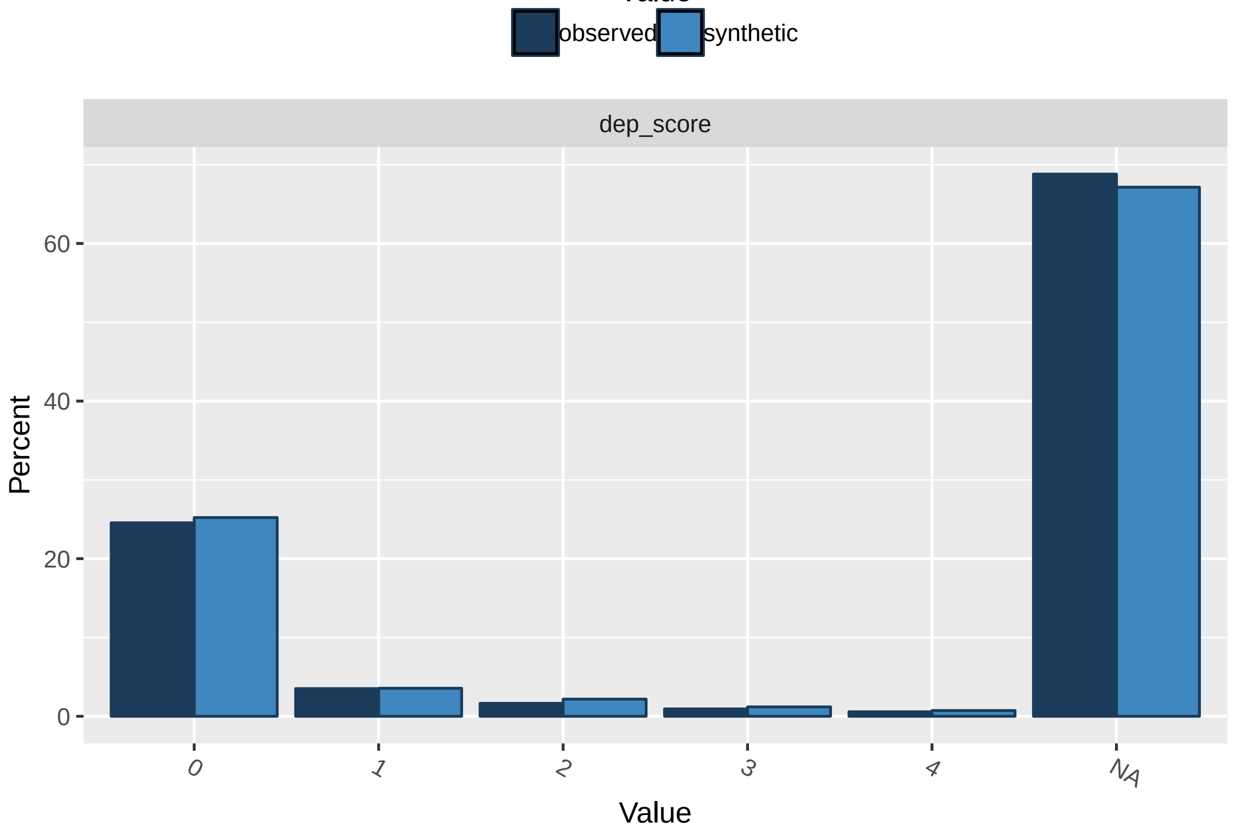

observed
synthetic
has_dep_diag
Percent
60
40
20
0
No ICD−10 diagnosis of depression
Yes ICD−10 diagnosis of depression
NA
Value

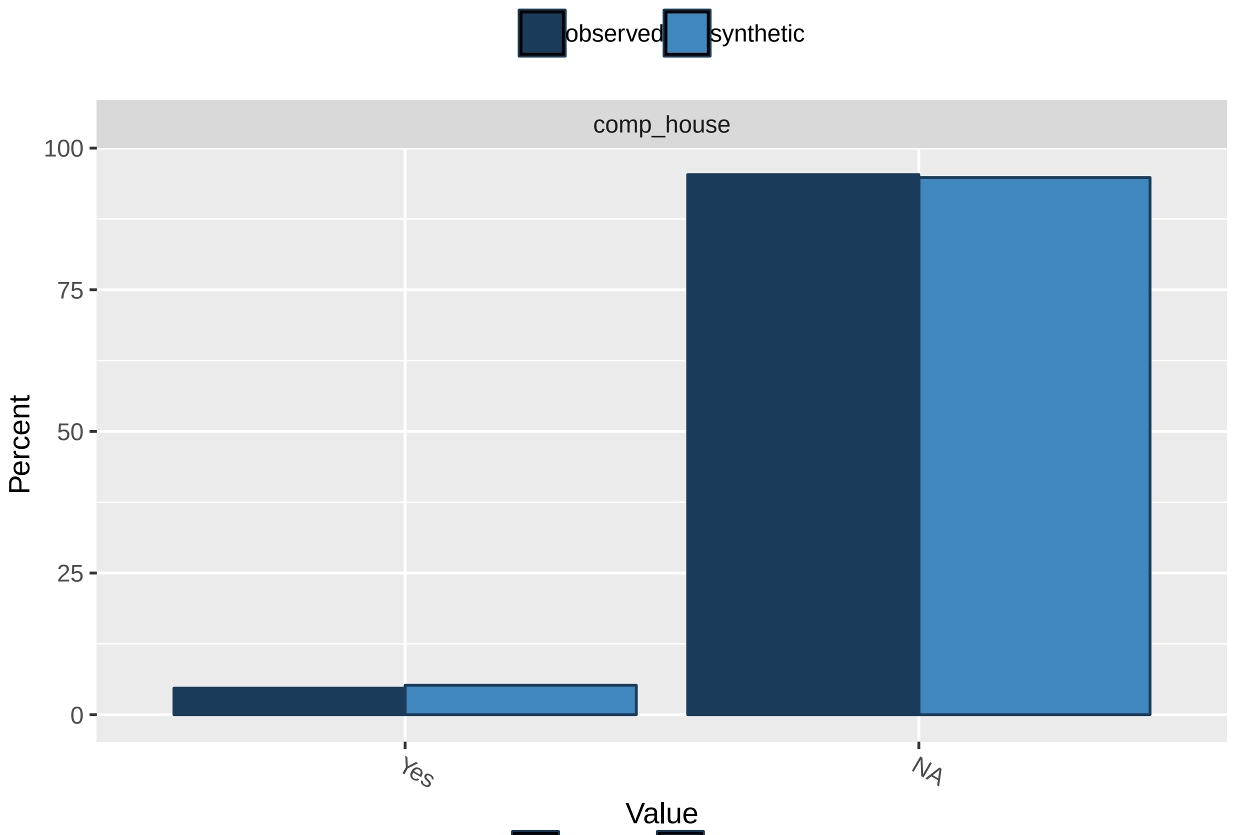

observed
synthetic
secd_diag
Percent
60
40
20
0
0
1
2
4
5
6
7
8
9
NA
Value

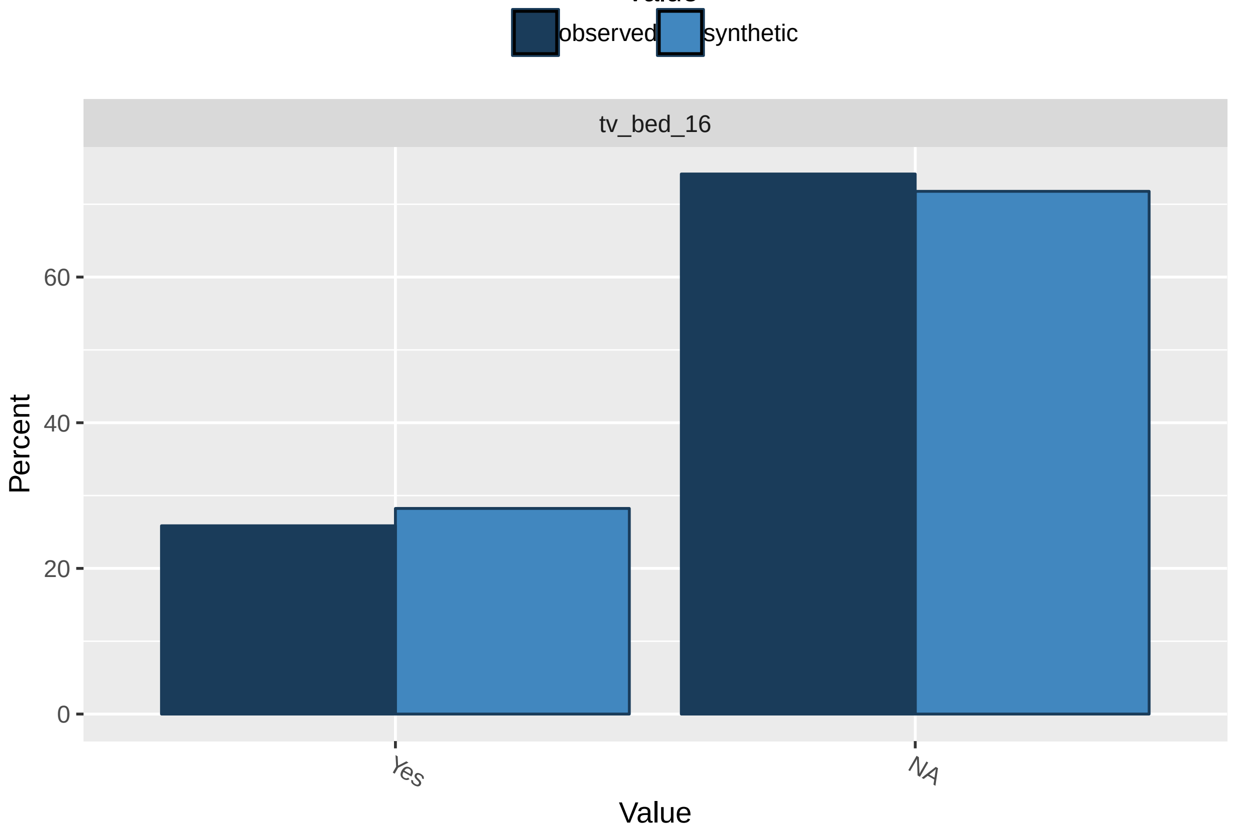

observed
synthetic
prim_diag
Percent
60
40
20
0
0
1
2
4
5
6
7
8
9
10
11
12
NA
Value

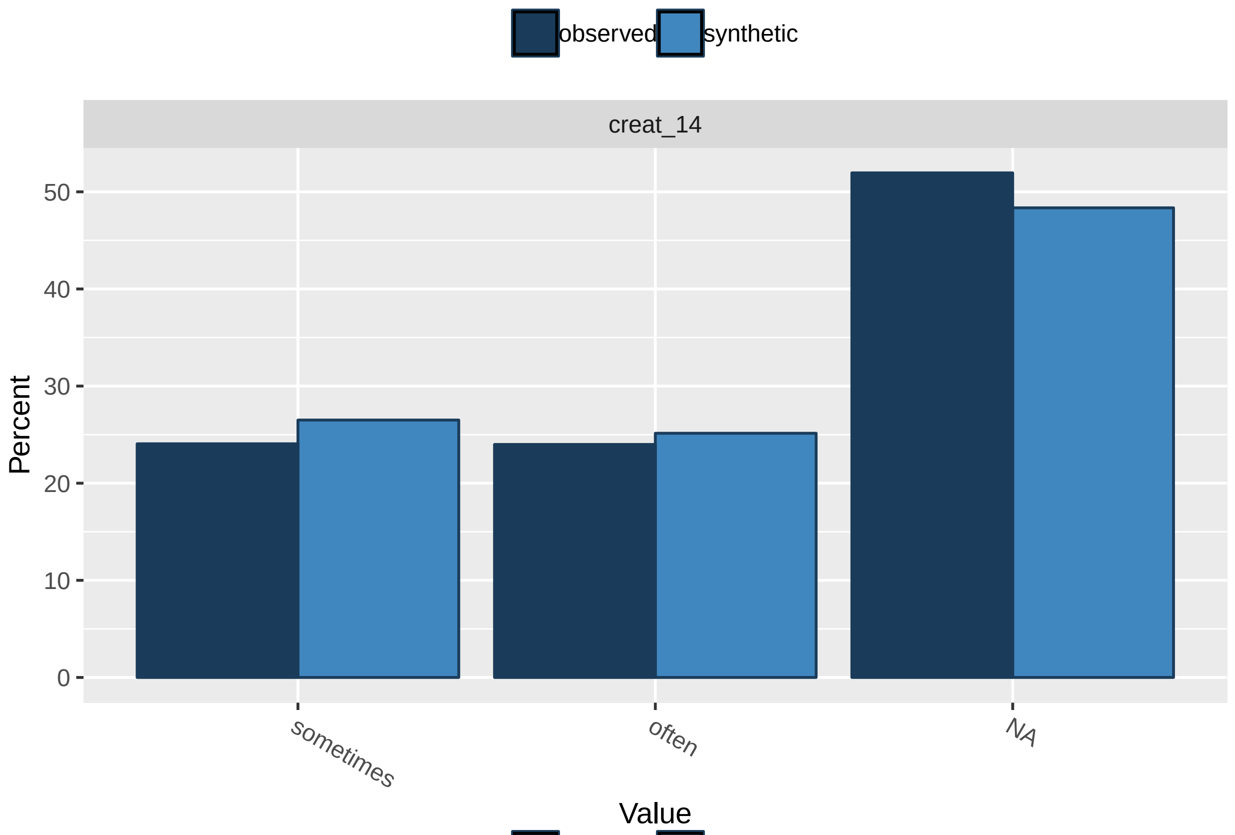

observed
synthetic
panic_score
Percent
60
40
20
0
0
1
2
3
4
NA
Value

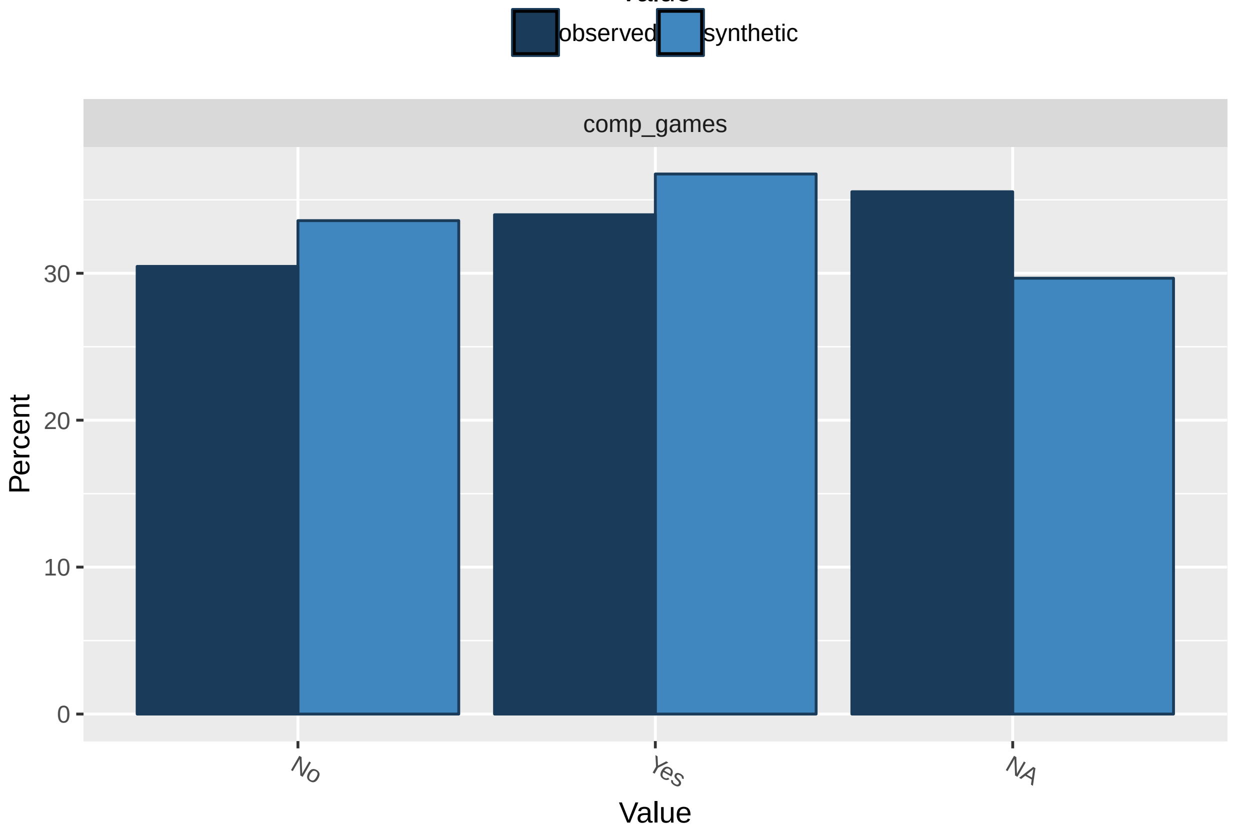

observed
synthetic
dep_thoughts
Percent
75
50
25
0
0
1
2
3
4
5
NA
Value

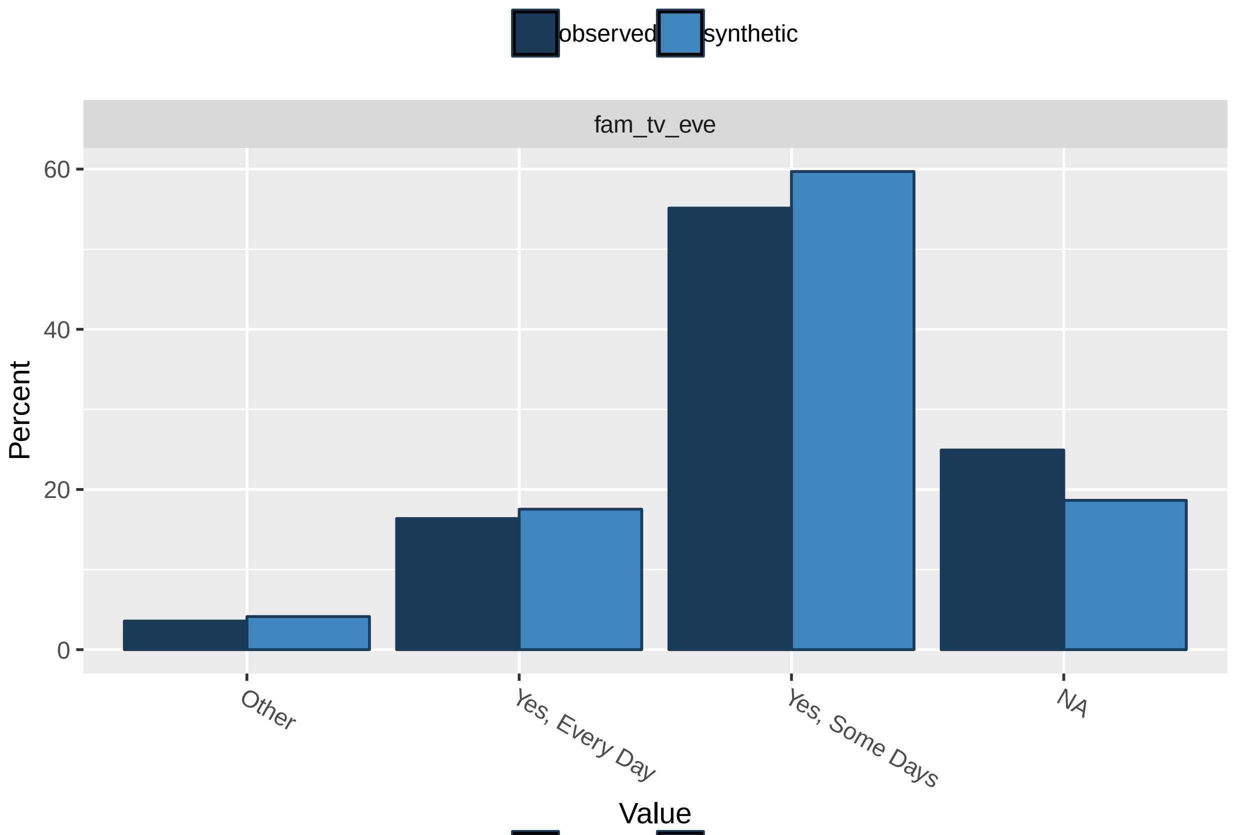

observed
synthetic
dep_score
Percent
60
40
20
0
0
1
2
3
4
NA
Value

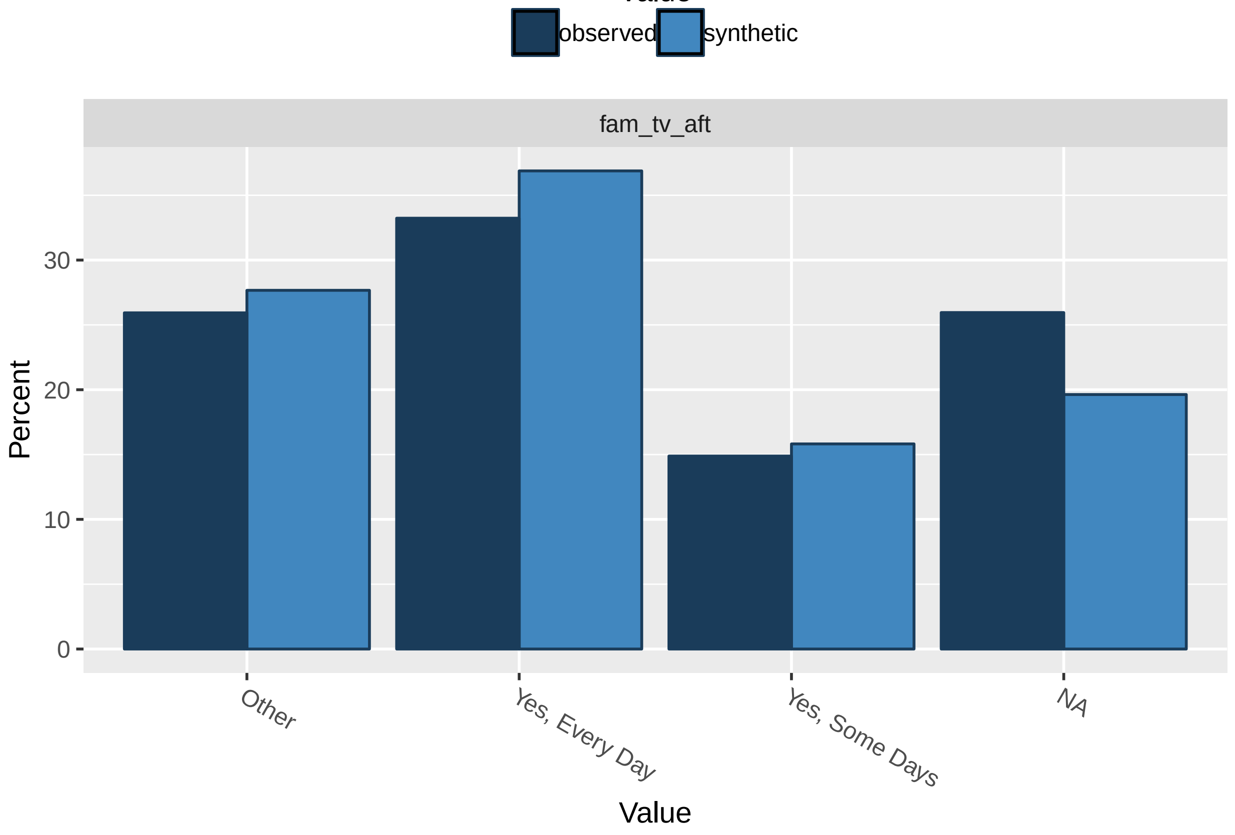

observed
synthetic
comp_house
Percent
100
75
50
25
0
Yes
NA
Value

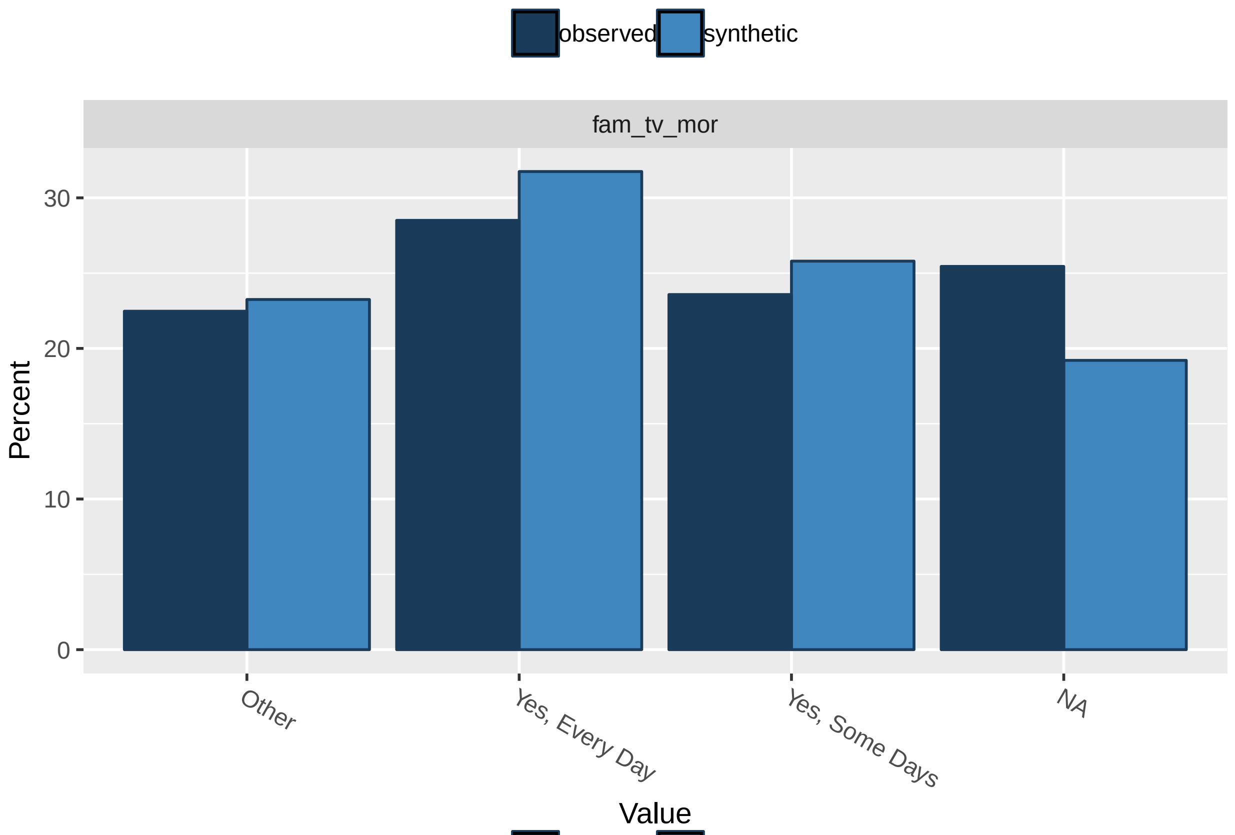

observed
synthetic
tv_bed_16
Percent
60
40
20
0
Yes
NA
Value


observed
synthetic
creat_14
Percent
0
10
20
30
40
50
sometimes
often
NA
Value

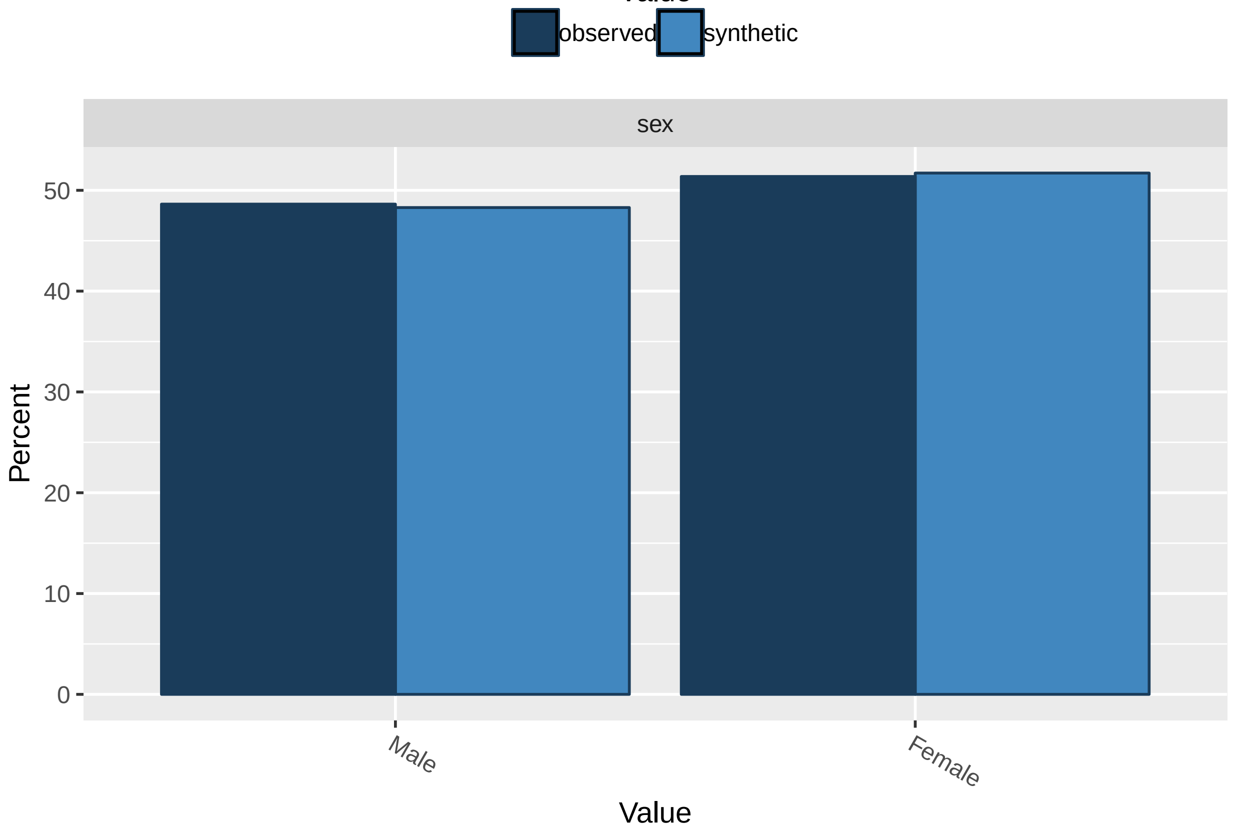

observed
synthetic
comp_games
Percent
0
10
20
30
No
Yes
NA
Value


observed
synthetic
fam_tv_eve
Percent
0
20
40
60
Other
Yes, Every Day
Yes, Some Days
NA
Value

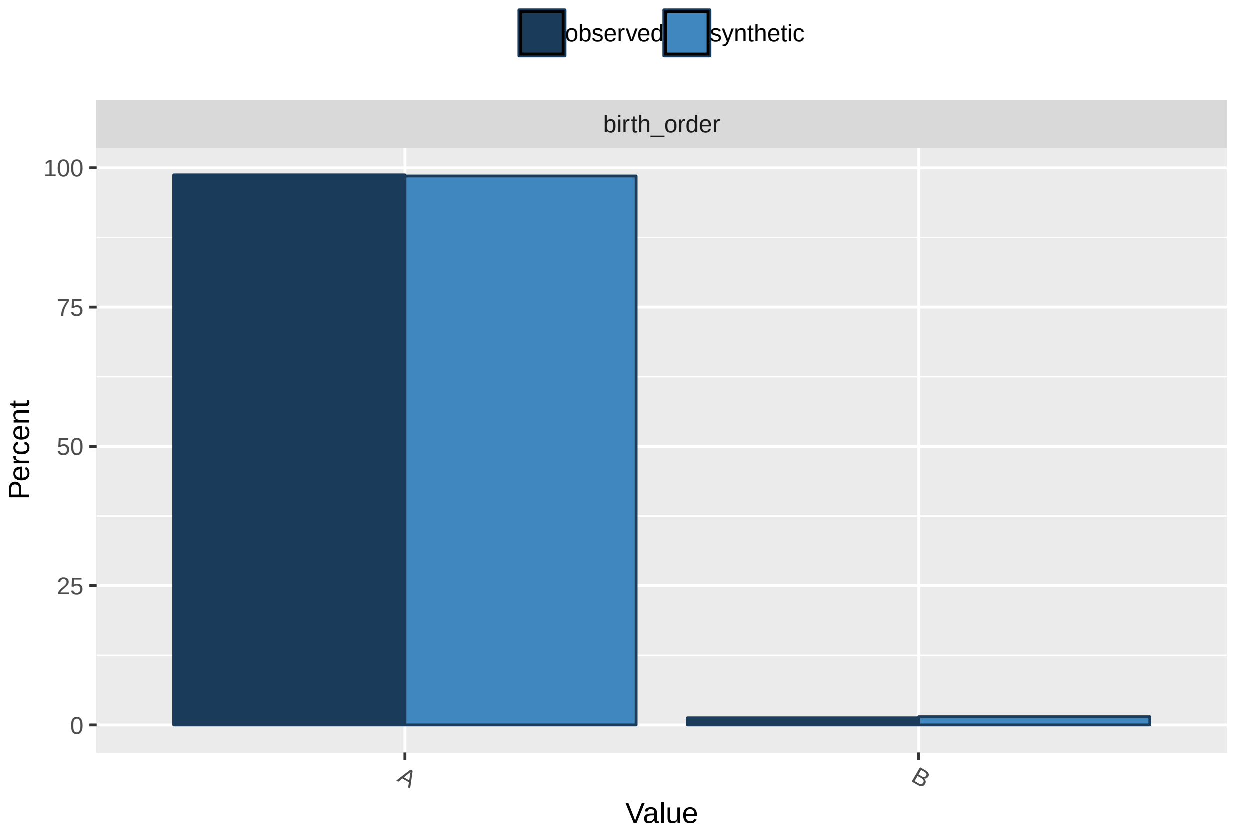

observed
synthetic
fam_tv_aft
Percent
0
10
20
30
Other
Yes, Every Day
Yes, Some Days
NA
Value



observed
synthetic
fam_tv_mor
Percent
0
10
20
30
Other
Yes, Every Day
Yes, Some Days
NA
Value



observed
synthetic
sex
Percent
0
10
20
30
40
50
Male
Female
Value



Write synthetic data:

Finally we'll double check the data doesn't contain any unique rows from the original dataset, add a column indicating the data is synthetic and then write out as a csv.

To double check we can use the SynthPop function to remove any unique duplicates.

```
sds <- sdc(sds, ods, label='synthetic',  rm.replicated.uniques = T)
```

```
## no. of replicated uniques: 0
```

```
dim(sds$syn)
```

```
## [1] 13734    83
```

```
write.csv( sds$syn, 'output/maps-synthetic-data.csv')
```