Testing different imputation methods on PUMS (MCAR) - Generative Adversarial Imputation Nets (GAIN)

```
# load dataset: df
load('../Datasets/ordinalPUMS.Rdata')

# take 10,000 samples: df
set.seed(0)
n = 10000
sample <- sample(nrow(df), size = 10000)
df <- df[sample,]

# create MCAR scneario with 30% chance of missing: df_observed
missing_prob = 0.3
df_observed <- df
missing_col = colnames(df)[c(1,3,5,7,9,11)]
for (col in missing_col) {
  missing_ind <- rbernoulli(n,p = missing_prob)
  df_observed[missing_ind, col] <- NA
}
```

Generative Adversarial Imputation Nets (GAIN)

reference: https://arxiv.org/abs/1806.02920

```
# Load imputed dataset
d1 = read.csv('../GAIN/imputed_dataset/PUMS_test_1.csv', header = FALSE, sep = ',')
d2 = read.csv('../GAIN/imputed_dataset/PUMS_test_2.csv', header = FALSE, sep = ',')
d3 = read.csv('../GAIN/imputed_dataset/PUMS_test_3.csv', header = FALSE, sep = ',')
d4 = read.csv('../GAIN/imputed_dataset/PUMS_test_4.csv', header = FALSE, sep = ',')
d5 = read.csv('../GAIN/imputed_dataset/PUMS_test_5.csv', header = FALSE, sep = ',')
colnames(d1) = colnames(df)
colnames(d2) = colnames(df)
colnames(d3) = colnames(df)
colnames(d4) = colnames(df)
colnames(d5) = colnames(df)
imputed_df = rbind(d1, d2, d3, d4, d5)
```

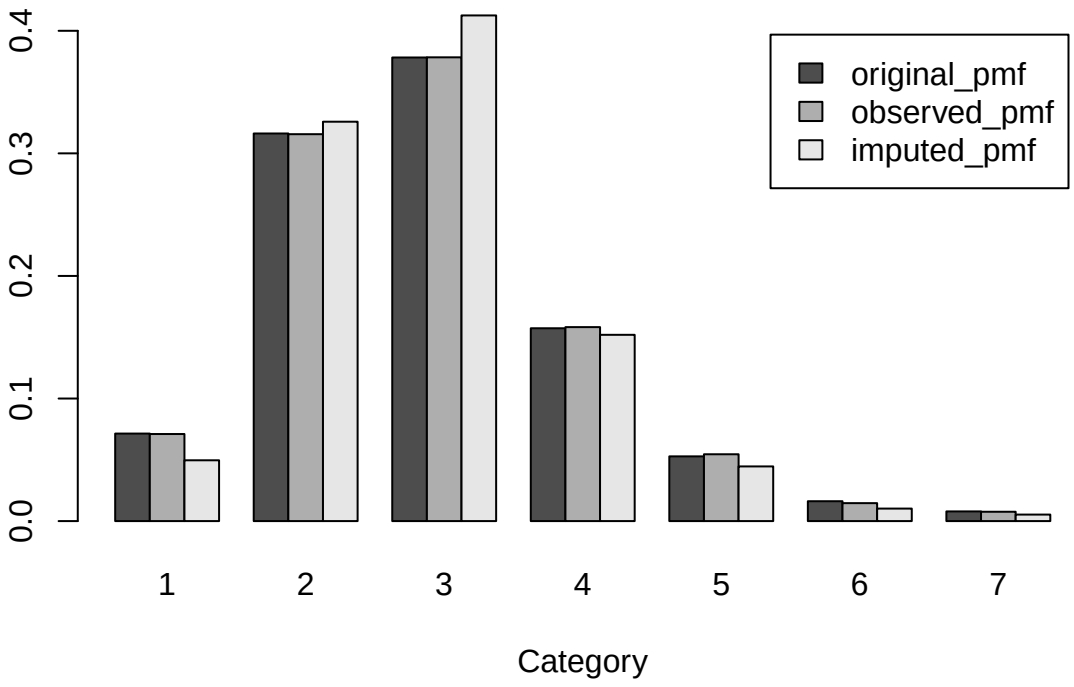
Diagnostics

Assess bivariate joint distribution

Assess bivariate joint distribution

```
# calculate rmse
numeric_df = sapply(df, as.numeric)
normalized_df = t(t(numeric_df-1)/(apply(numeric_df, MARGIN = 2, FUN = max)-1))
numeric_impute = sapply(d1, as.numeric)
normalized_impute = t(t(numeric_impute-1)/(apply(numeric_df, MARGIN = 2, FUN = max)-1))

missing_matrix = is.na(df_observed)
```



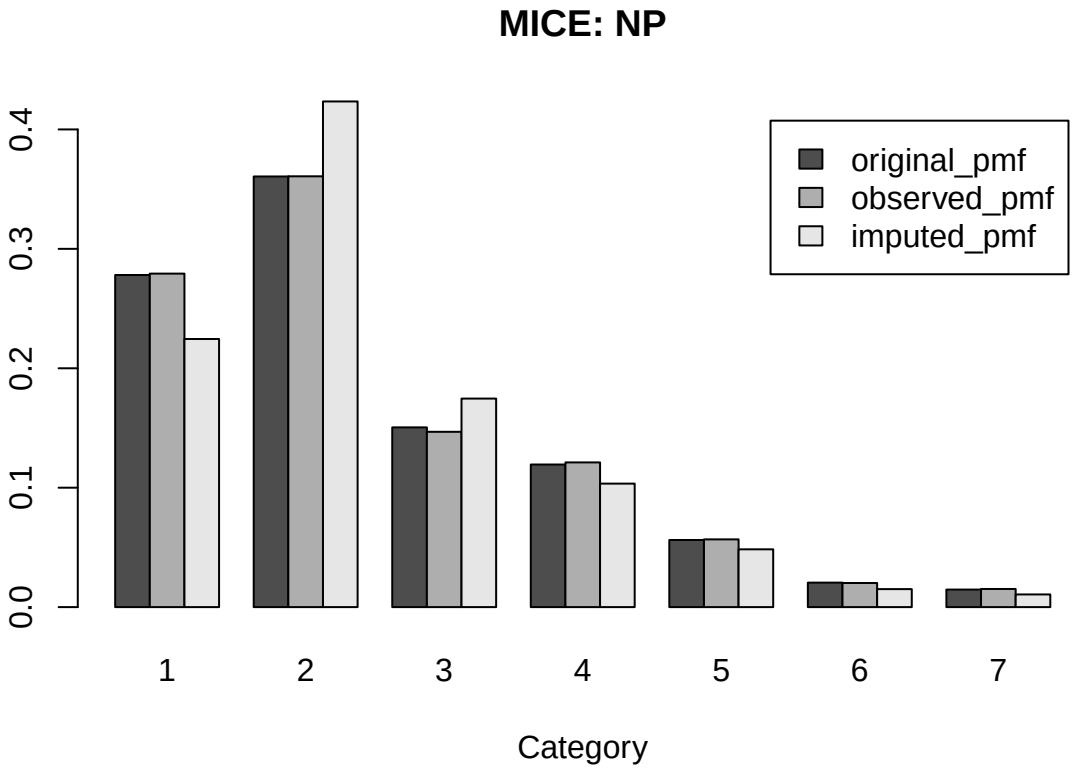

```
rmse = sqrt(sum((normalized_df[missing_matrix] - normalized_impute[missing_matrix])^2)/sum(missing_matri
rmse
```

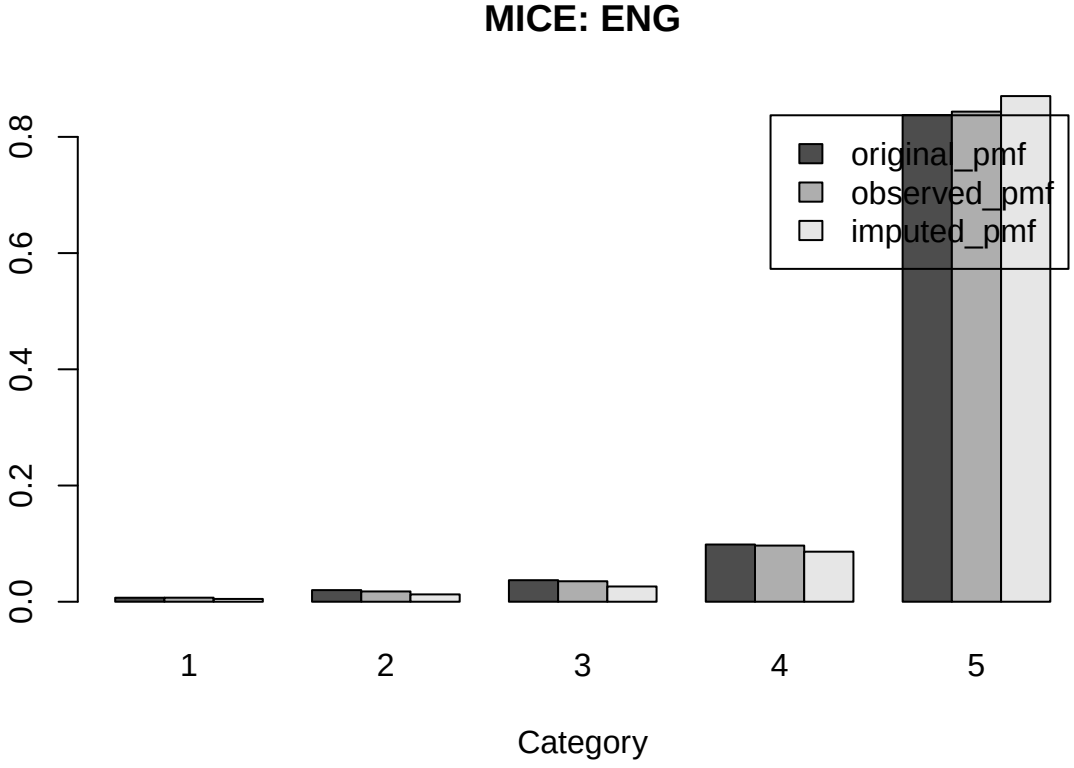
```
## [1] 0.1900414
```

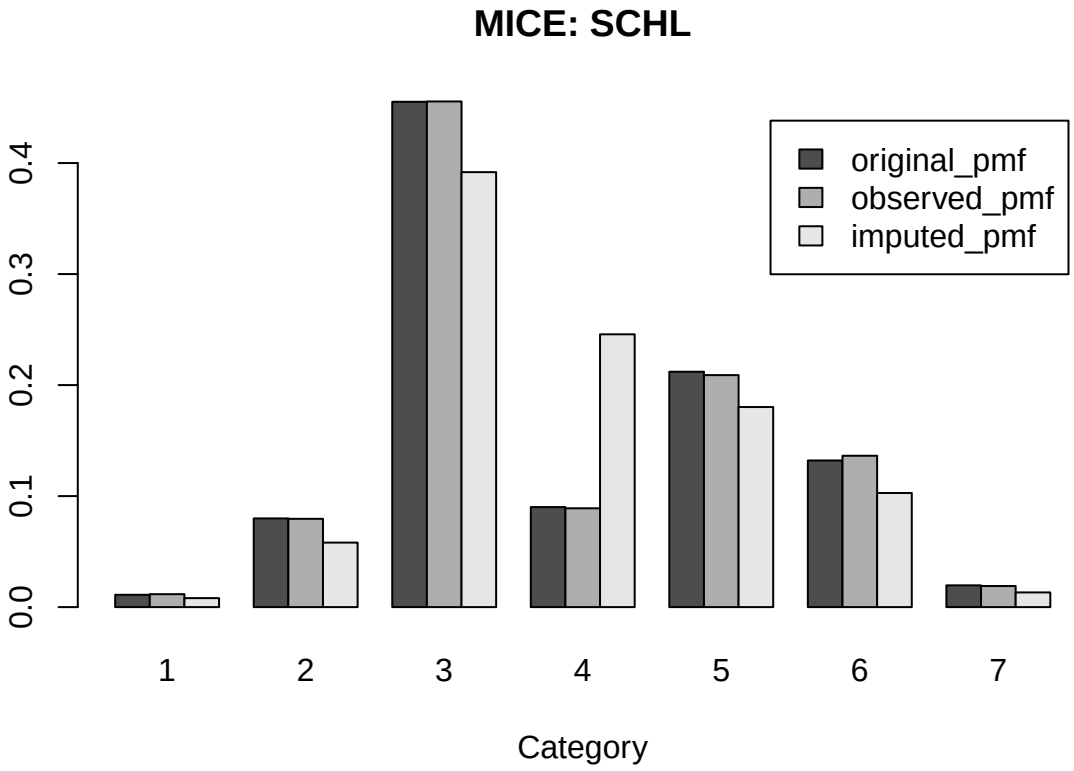
```
# accuracy
acc = sum(numeric_df[missing_matrix] == numeric_impute[missing_matrix])/sum(missing_matrix)
acc
```

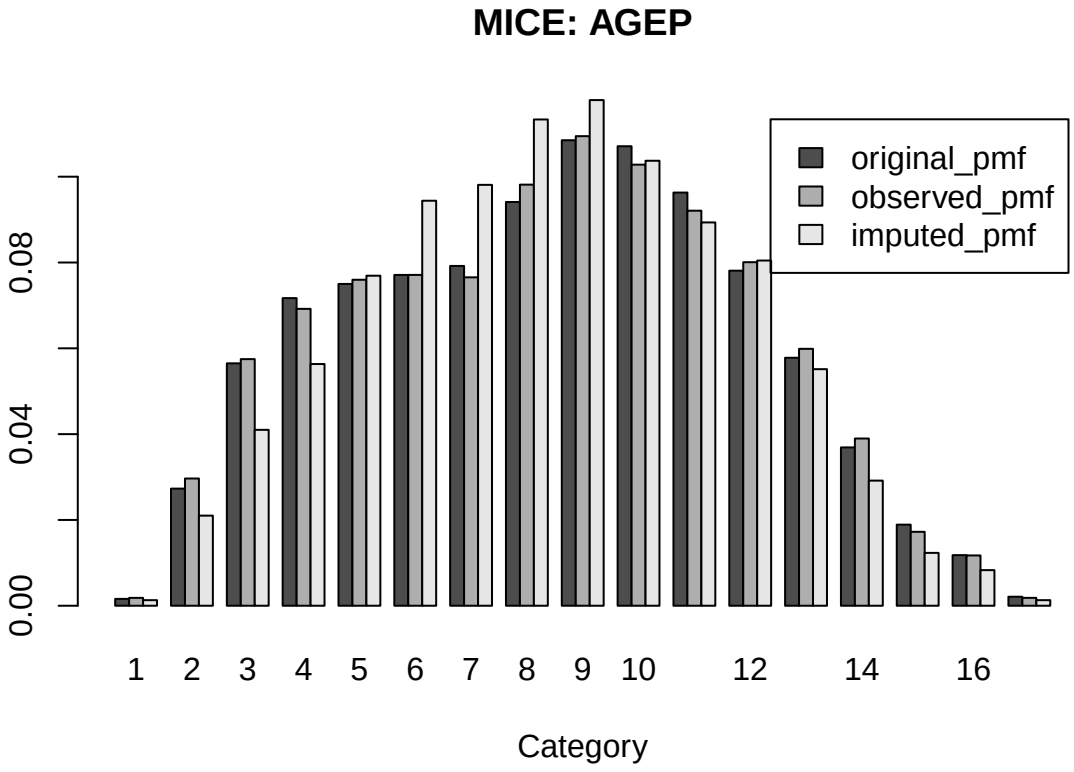
```
## [1] 0.3872108
```

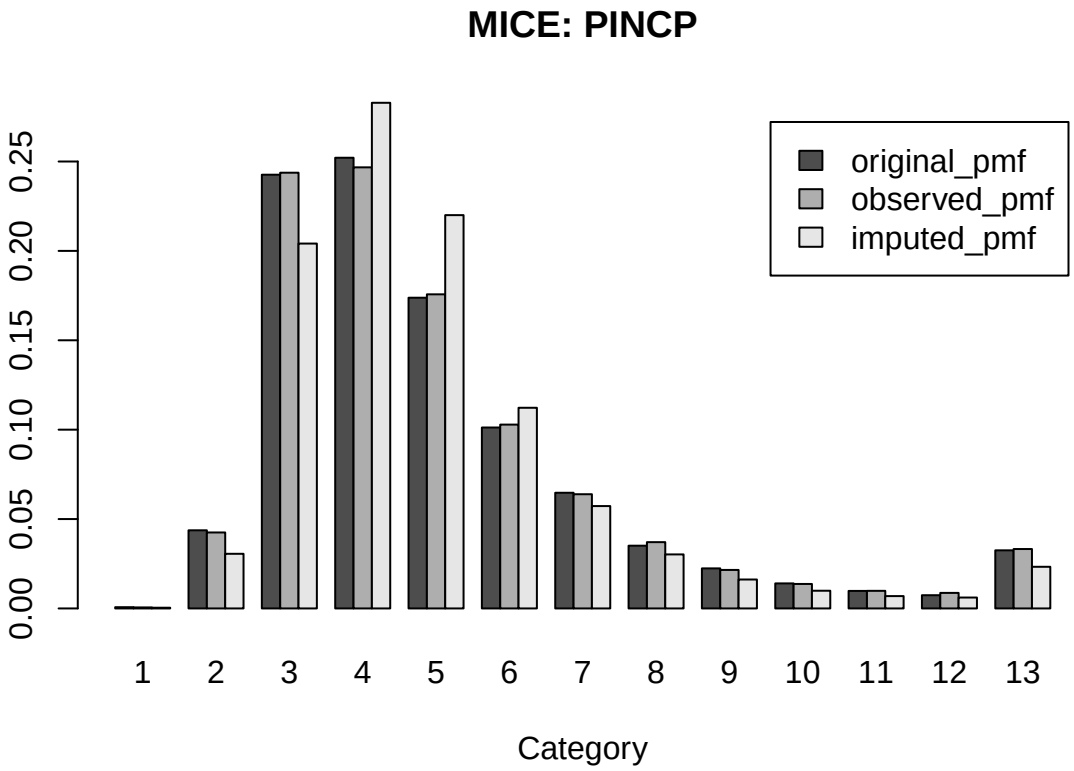
```
# Actual vs Imputed values
col = 1
missing_indicator = missing_matrix[,col]
plot(as.integer(df[missing_indicator, col]), as.integer(d1[missing_indicator, col]), xlab = 'actual valu
```

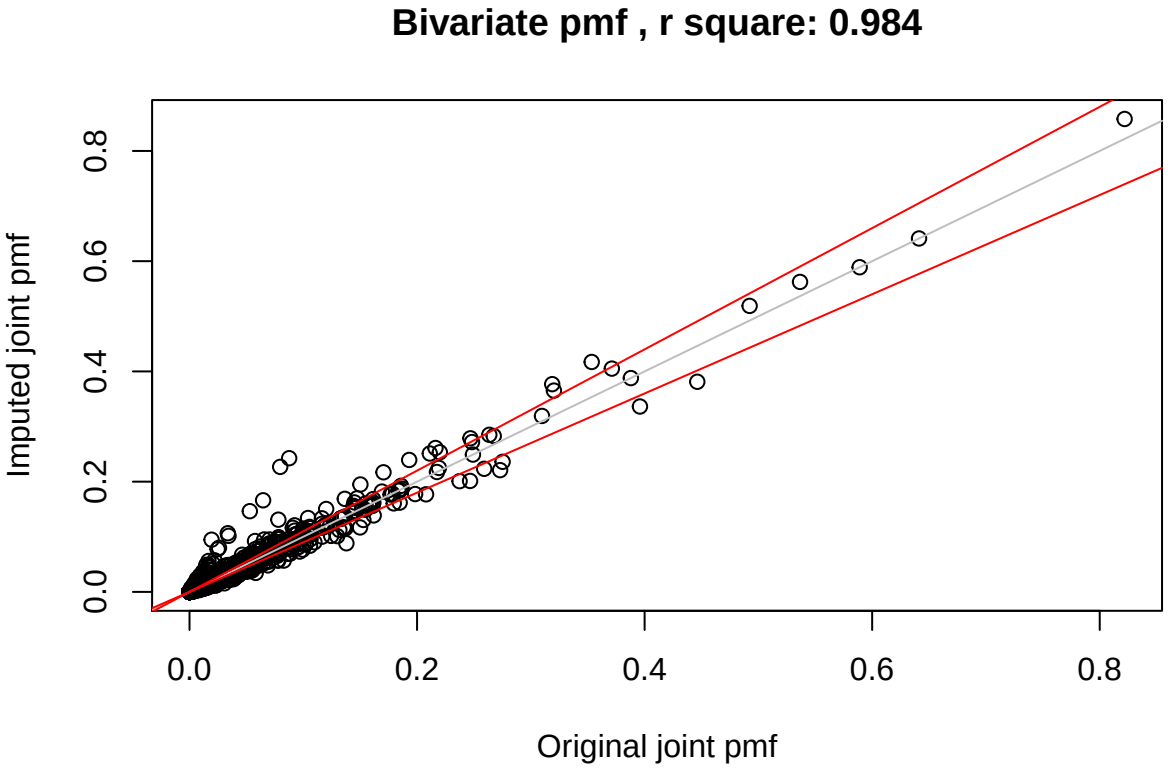

MICE: NP
0.0
0.1
0.2
0.3
0.4
original_pmf
observed_pmf
imputed_pmf
1
2
3
4
5
6
7
Category

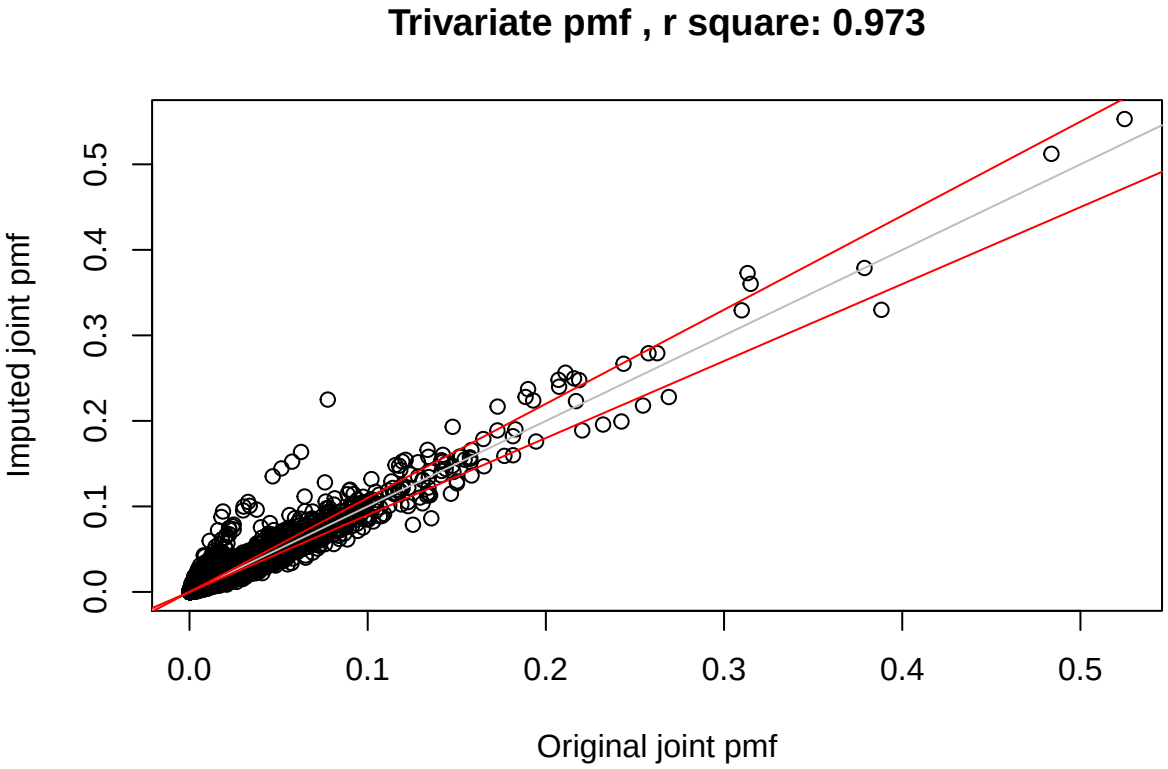

MICE: ENG
0.0
0.2
0.4
0.6
0.8
original_pmf
observed_pmf
imputed_pmf
1
2
3
4
5
Category

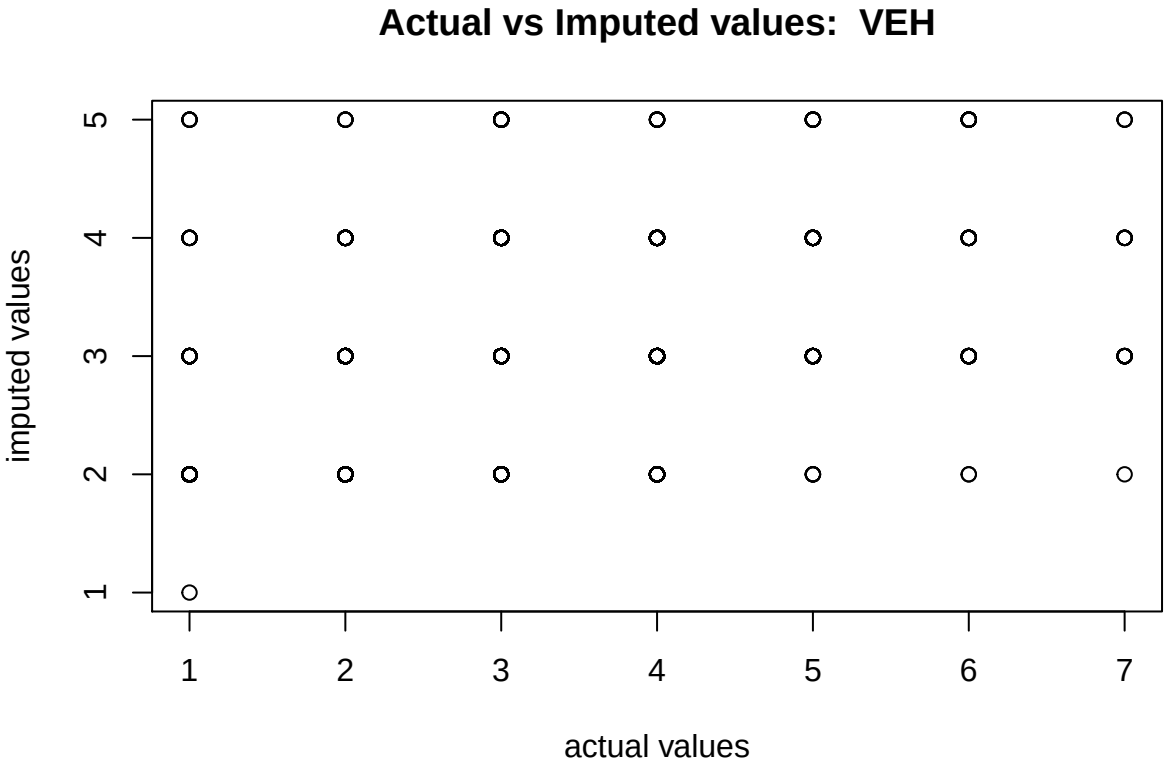

MICE: SCHL
original_pmf
observed_pmf
imputed_pmf
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
Category


MICE: AGEP
original_pmf
observed_pmf
imputed_pmf
0.00
0.04
0.08
1
2
3
4
5
6
7
8
9
10
12
14
16
Category


MICE: PINCP
original_pmf
observed_pmf
imputed_pmf
0.00
0.05
0.10
0.15
0.20
0.25
1
2
3
4
5
6
7
8
9
10
11
12
13
Category


Bivariate pmf , r square: 0.984
Imputed joint pmf
0.0
0.2
0.4
0.6
0.8
Original joint pmf


Trivariate pmf , r square: 0.973
Imputed joint pmf
0.0
0.1
0.2
0.3
0.4
0.5
Original joint pmf


Actual vs Imputed values: VEH
imputed values
1
2
3
4
5
6
7
actual values