Testing different imputation methods on PUMS (MAR) - MICE-CART

```r
# 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 = c(1,3,7,9,10,11)

# Make VEH and WKL MCAR
missing_col_MCAR = c(1,10)
for (col in missing_col_MCAR) {
  missing_ind <- rbernoulli(n,p = missing_prob)
  df_observed[missing_ind, col] <- NA
}

# Make the rest MAR
numeric_df = sapply(df, as.numeric)
normalized_df = t(t(numeric_df-1)/(apply(numeric_df, MARGIN = 2, FUN = max)-1))
missing_col_MAR = c(3,7,9,11)
fully_observed_col = c(2,4,5,6,8)

cutoff_NP = c(0, 0.2, 0.3, 0.5, 0)
weight_NP = c(0, 0.3, -0.2, 0.15, 0)
beta0_NP = 0.05

cutoff_SCHL = c(0.5, 0.4, 0.8, 0, 0)
weight_SCHL = c(0.1, -0.3, 0.02, 0, 0)
beta0_SCHL = 0.05

cutoff_AGEP = c(0.4, 0.3, 0, 0.2, 0)
weight_AGEP = c(0.1, 0.4, 0, -0.2, 0)
beta0_AGEP = 0.05

cutoff_PINCP = c(0.7, 0, 0.6, 0.5, 0)
weight_PINCP = c(-0.25, 0, 0.02, 0.1, 0)
beta0_PINCP = 0.05

# missing probability for NP
prob_NP = apply(t((t(normalized_df[, fully_observed_col]) > cutoff_NP)*weight_NP)
                  , MARGIN = 1, sum)
prob_NP = prob_NP-min(prob_NP) + beta0_NP
```

```
indicator = rbernoulli(n, p = prob_NP)
df_observed[indicator, missing_col_MAR[1]] <- NA

# missing probability for SCHL
prob_SCHL = apply(t((t(normalized_df[, fully_observed_col]) > cutoff_SCHL)*weight_SCHL)
                  , MARGIN = 1, sum)
prob_SCHL = prob_SCHL-min(prob_SCHL) + beta0_SCHL
indicator = rbernoulli(n, p = prob_SCHL)
df_observed[indicator, missing_col_MAR[2]] <- NA

# missing probability for AGEP
prob_AGEP = apply(t((t(normalized_df[, fully_observed_col]) > cutoff_AGEP)*weight_AGEP)
                  , MARGIN = 1, sum)
prob_AGEP = prob_AGEP - min(prob_AGEP) + beta0_AGEP
indicator = rbernoulli(n, p = prob_AGEP)
df_observed[indicator, missing_col_MAR[3]] <- NA

# missing probability for PINCP
prob_PINCP = apply(t((t(normalized_df[, fully_observed_col]) > cutoff_PINCP)*weight_PINCP)
                  , MARGIN = 1, sum)
prob_PINCP = prob_PINCP - min(prob_PINCP) + beta0_PINCP
indicator = rbernoulli(n, p = prob_PINCP)
df_observed[indicator, missing_col_MAR[4]] <- NA

# 30.80% missing
apply(is.na(df_observed), MARGIN = 2, mean)
```

```
##    VEH     MV     NP   RMSP    ENG  MARHT   SCHL RACNUM   AGEP    WKL  PINCP
## 0.3030 0.0000 0.3183 0.0000 0.0000 0.0000 0.3497 0.0000 0.2946 0.3017 0.2809
```

```
unique(prob_NP)
```

```
## [1] 0.35 0.20 0.05 0.50 0.25 0.55 0.70 0.40
```

```
unique(prob_SCHL)
```

```
## [1] 0.07 0.37 0.47 0.17 0.45 0.35 0.15 0.05
```

```
unique(prob_AGEP)
```

```
## [1] 0.45 0.15 0.55 0.05 0.25 0.65 0.35 0.75
```

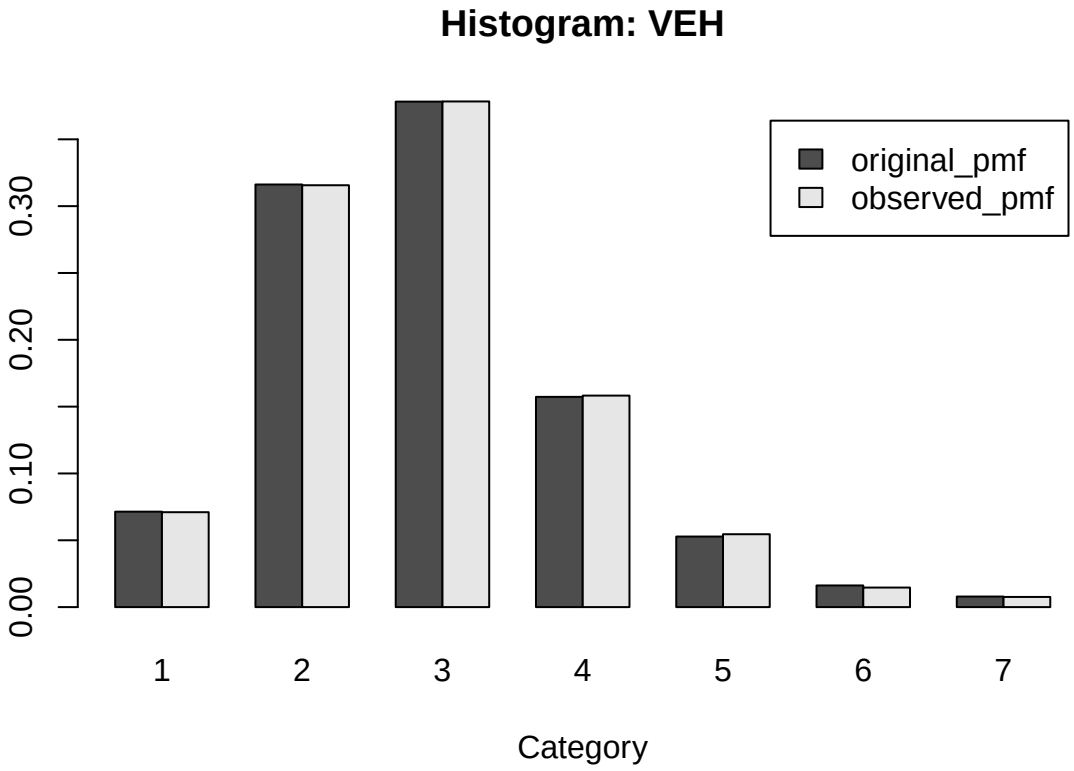
```
unique(prob_PINCP)
```

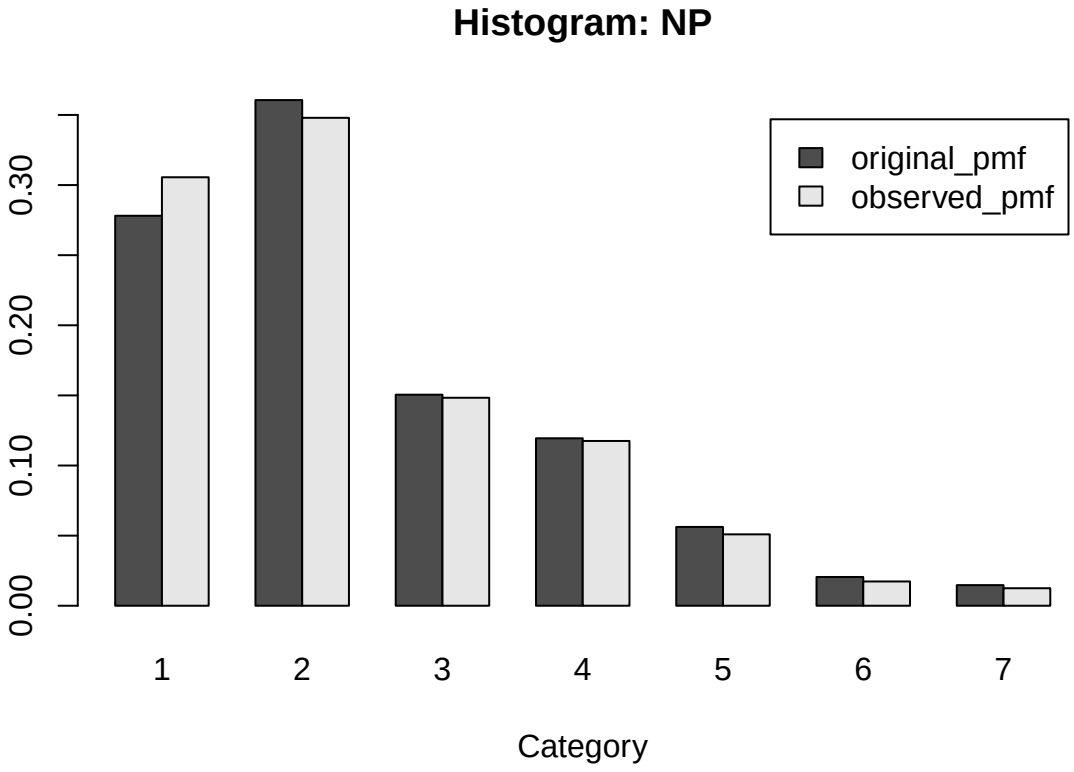
```
## [1] 0.32 0.42 0.07 0.17 0.30 0.40 0.05 0.15
```

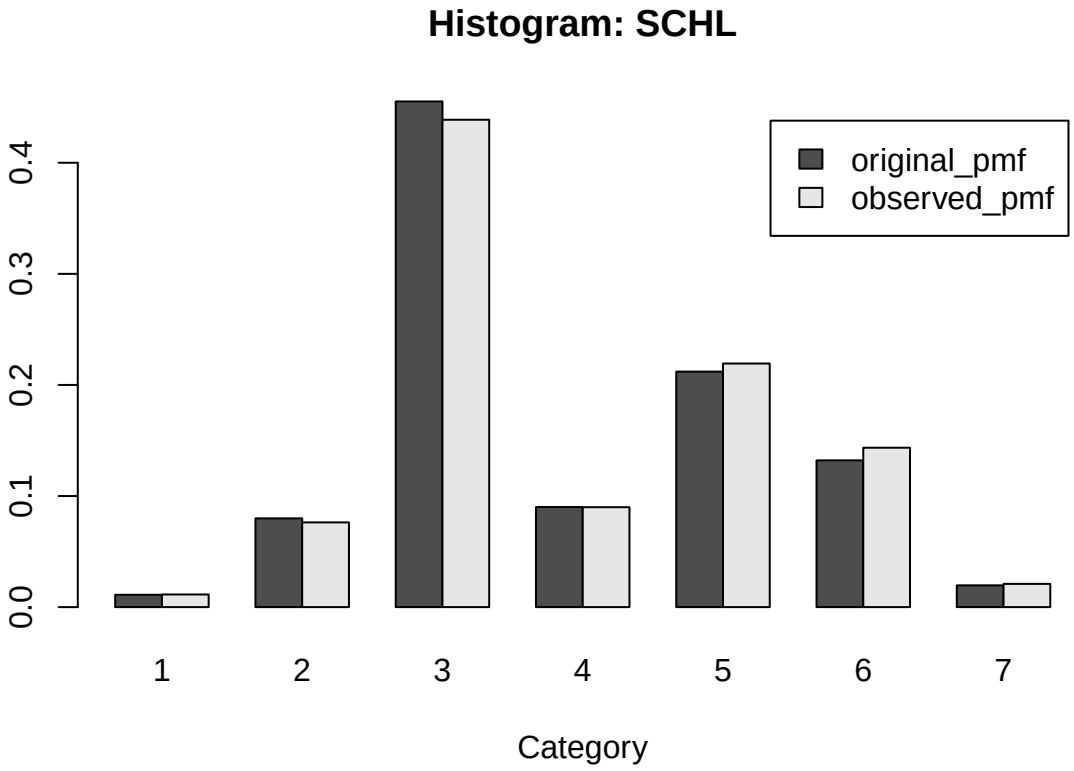
Histogram for univariate distribution

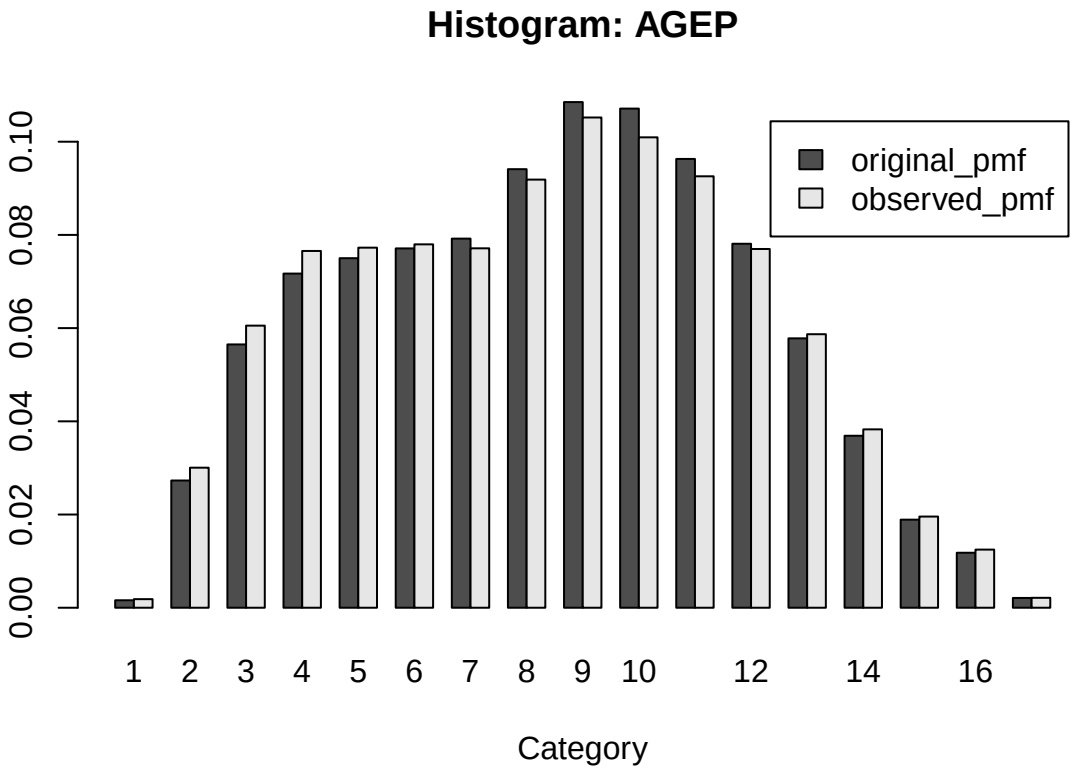
Assess bivariate joint distribution

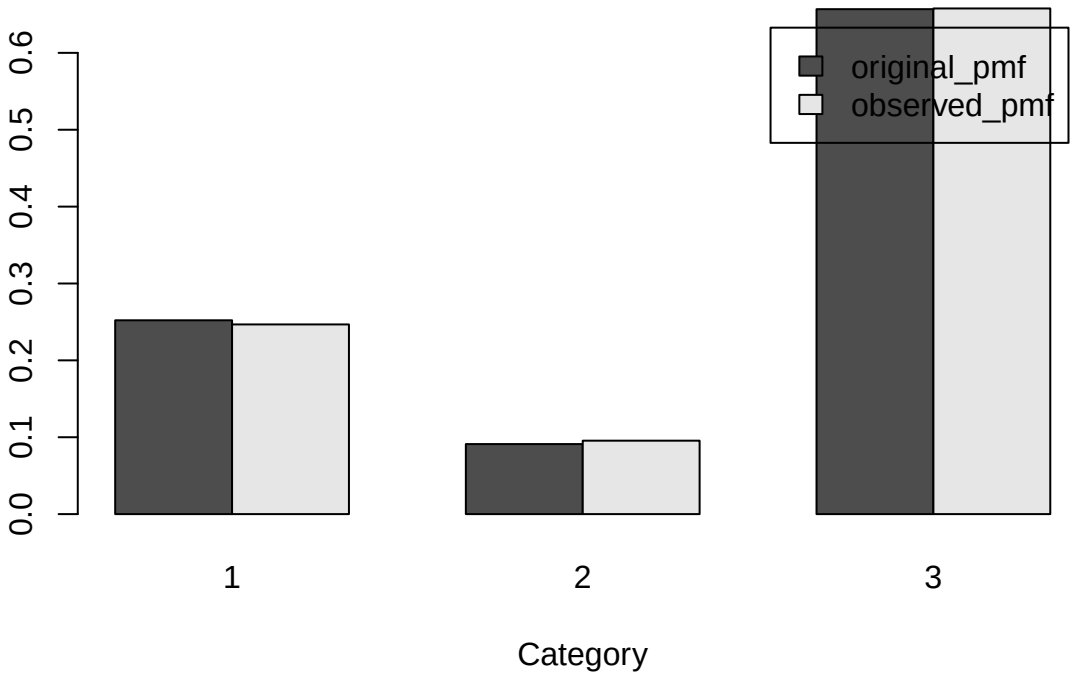
Assess trivariate joint distribution

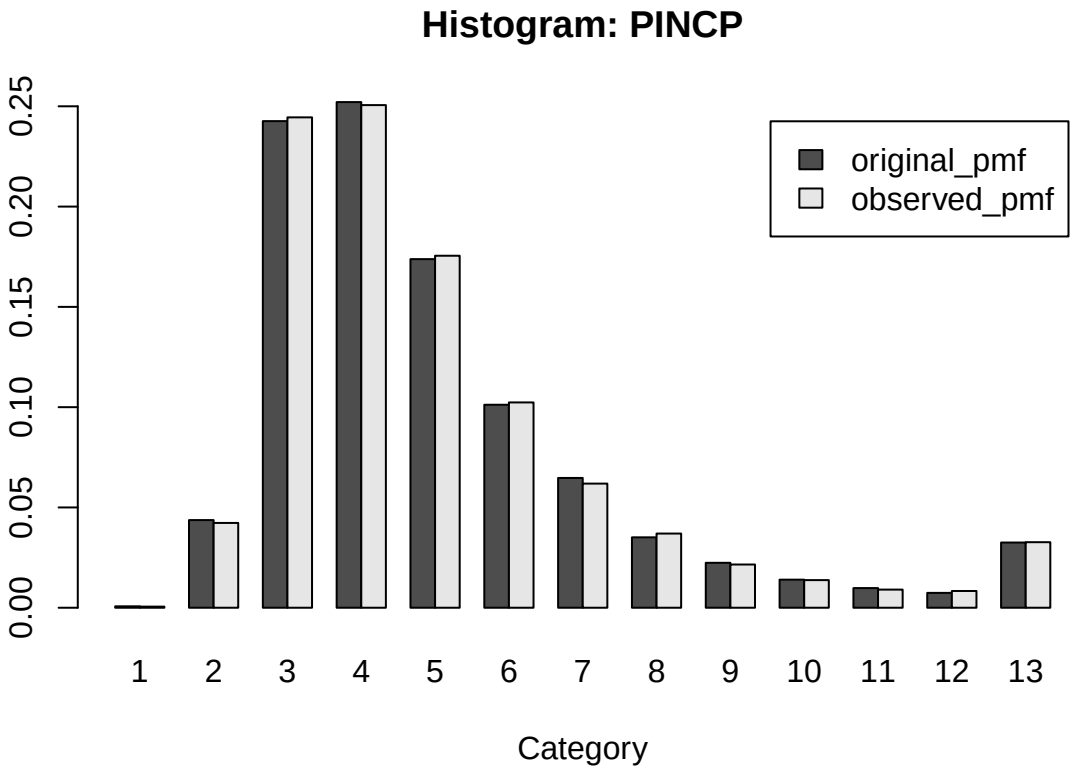

Histogram: VEH
original_pmf
observed_pmf
0.00
0.10
0.20
0.30
1
2
3
4
5
6
7
Category

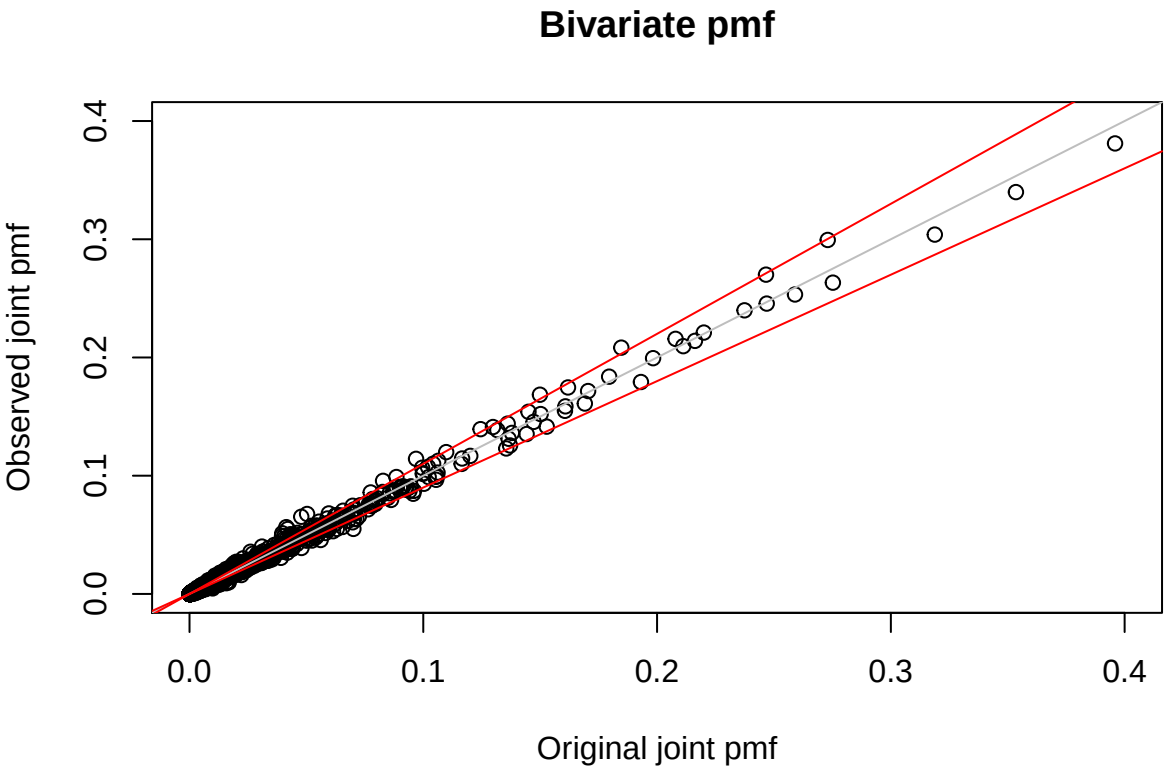

Histogram: NP
original_pmf
observed_pmf
0.00
0.10
0.20
0.30
1
2
3
4
5
6
7
Category

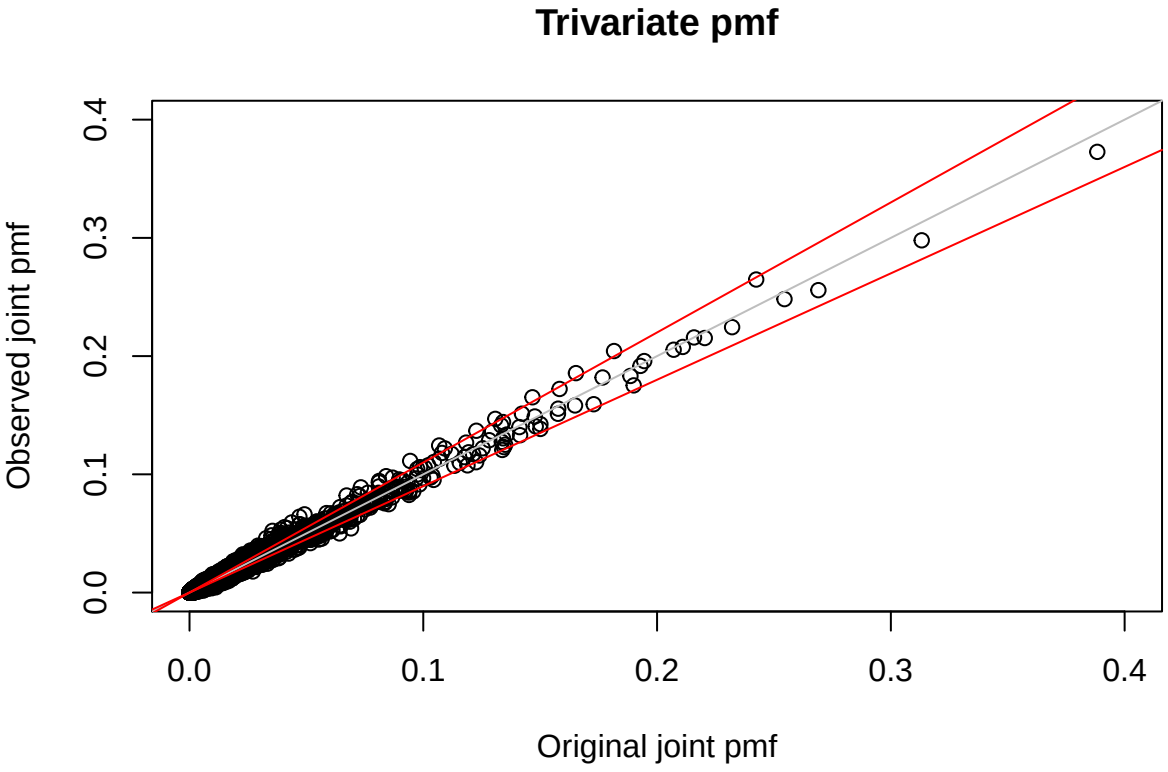

Histogram: SCHL
original_pmf
observed_pmf
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
Category


Histogram: AGEP
original_pmf
observed_pmf
0.00
0.02
0.04
0.06
0.08
0.10
1
2
3
4
5
6
7
8
9
10
12
14
16
Category


Histogram: WKL
original_pmf
observed_pmf
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1
2
3
Category


Histogram: PINCP
original_pmf
observed_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
Observed joint pmf
0.0
0.1
0.2
0.3
0.4
Original joint pmf


Trivariate pmf
Observed joint pmf
0.0
0.1
0.2
0.3
0.4
Original joint pmf