Testing different imputation methods on PUMS (MAR) - DPMPM

```
# 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 45% chance of missing: df_observed
missing_prob = 0.45
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)
beta_NP = c(-0.05, -1.5, 0.6, -2, -0.05)+c(0,0.45,0.45,0.45,0)
beta0_NP = -0.05
beta_SCHL = c(-3, 3, -0.75, 0.05, -0.2)+c(0.5,0.5,0.5,0,0)
beta0_SCHL = 0.05
beta_AGEP = c(0.05, -0.2, 0.05, -1.25, 1)+c(0,0,0,1.1,1.1)
beta0_AGEP = -0.05
beta_PINCP = c(3, -0.05, -2.5, 0.05, -1)+c(0.5,0,0.5,0,0.5)
beta0_PINCP = -0.05

# missing probability for NP
prob_NP = apply(t(t(normalized_df[, fully_observed_col])*beta_NP)+beta0_NP, MARGIN = 1, sum)
prob_NP = exp(prob_NP)/(exp(prob_NP)+1)
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])*beta_SCHL)+beta0_SCHL, MARGIN = 1, sum)
prob_SCHL = exp(prob_SCHL)/(exp(prob_SCHL)+1)
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])*beta_AGEP)+beta0_AGEP, MARGIN = 1, sum)
prob_AGEP = exp(prob_AGEP)/(exp(prob_AGEP)+1)
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])*beta_PINCP)+beta0_PINCP, MARGIN = 1, sum)
prob_PINCP = exp(prob_PINCP)/(exp(prob_PINCP)+1)
indicator = rbernoulli(n, p = prob_PINCP)
df_observed[indicator, missing_col_MAR[4]] <- NA

# 44.99% missing
apply(is.na(df_observed), MARGIN = 2, mean)
```

```
##    VEH     MV     NP   RMSP    ENG  MARHT   SCHL RACNUM   AGEP    WKL  PINCP
## 0.4554 0.0000 0.4645 0.0000 0.0000 0.0000 0.4465 0.0000 0.4328 0.4552 0.4454
```

DPMPM

Multiple imputation using NPBayesImputeCat package

Ref: https://cran.r-project.org/web/packages/NPBayesImputeCat/NPBayesImputeCat.pdf

1. Create and initialize the Rcpp_Lcm model object using CreateModel with the following arguments:

- X: dataframe to be imptuted = df
- MCZ: dataframe with the definition of structural zero = NULL
- K: the maximum number of mixture components = 40
- Nmax: An upper truncation limit for the augmented sample size = 0
- aalpha: the hyper parameter alpha in stick-breaking prior = 0.25
- balpha: the hyper parameter beta in stick-breaking prior = 0.25
- seed = 0

2. Set the tracer for the sampling process

- k_star: the effective cluster number
- psi: conditional multinomial probabilties
- ImputedX: imputation result

3. Run the model using the method Run of Rcpp_Lcm class with the following arguments:

- burnin = 10000
- iter = 10000
- thinning = 5

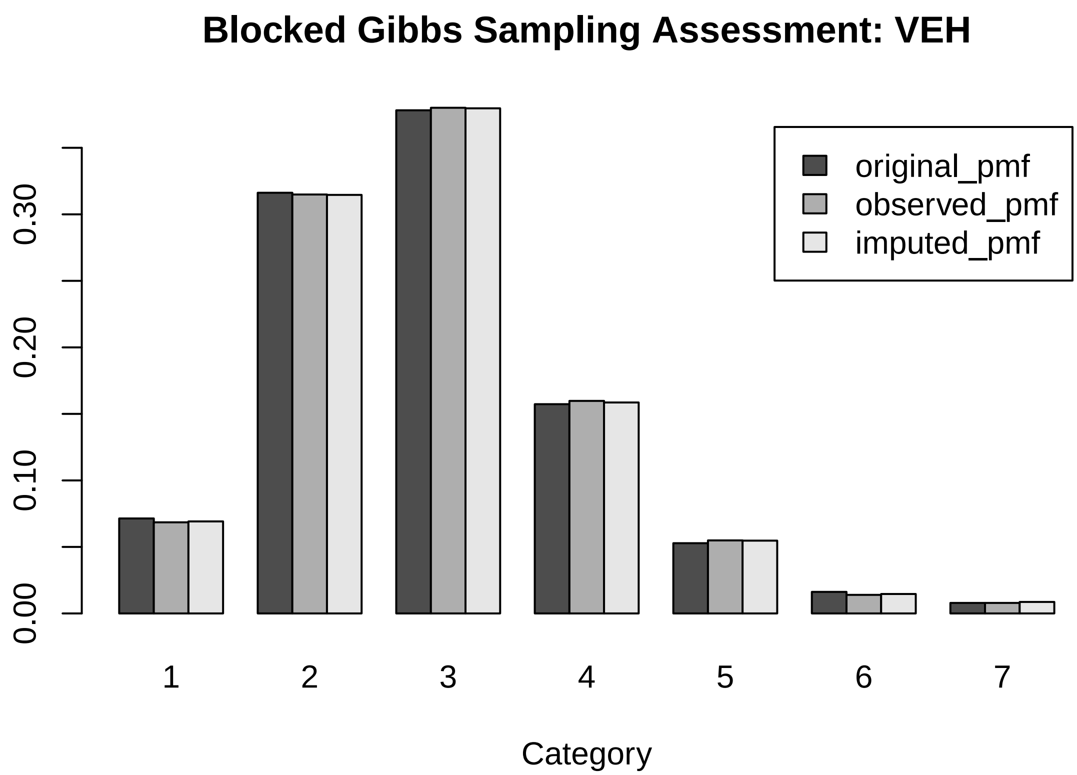
4. Obtain result

```
N = 40
Mon = 10000
B = 10000
thin.int = 5

# 1. Create and initialize the Rcpp_Lcm model object
model = CreateModel(X = df_observed, MCZ = NULL, K = N, Nmax = 0,
                    aalpha = 0.25, balpha = 0.25, seed = 0)
# 2. Set tracer
model$SetTrace(c('k_star', 'psi', 'ImputedX', 'alpha'),Mon)
```

```r
# 3. Run model using Run(burnin, iter, thinning)
model$Run(B,Mon,thin.int)
```

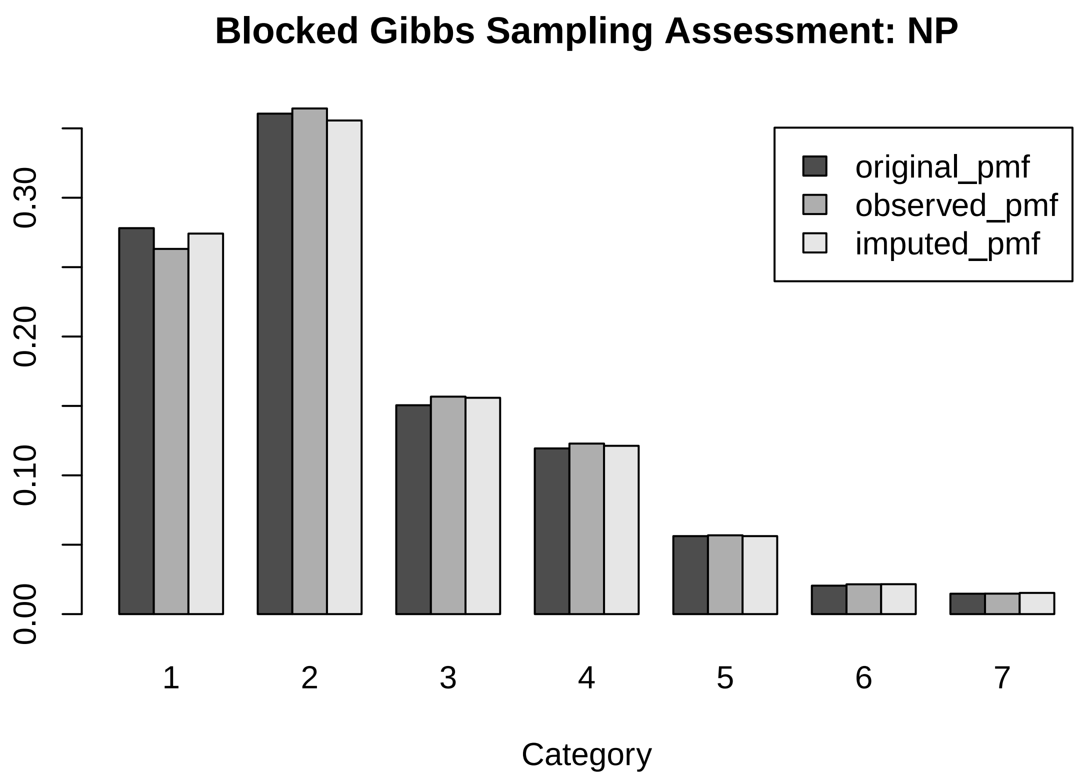
```r
# Extract results
output <- model$GetTrace()
k_star <- output$k_star
psi <- output$psi
imputed_df <- output$ImputedX
alpha <- output$alpha

#retrieve parameters from the final iteration
result <- model$snapshot

#convert ImputedX matrix to dataframe, using proper factors/names etc.
ImputedX <- GetDataFrame(result$ImputedX,df)
```

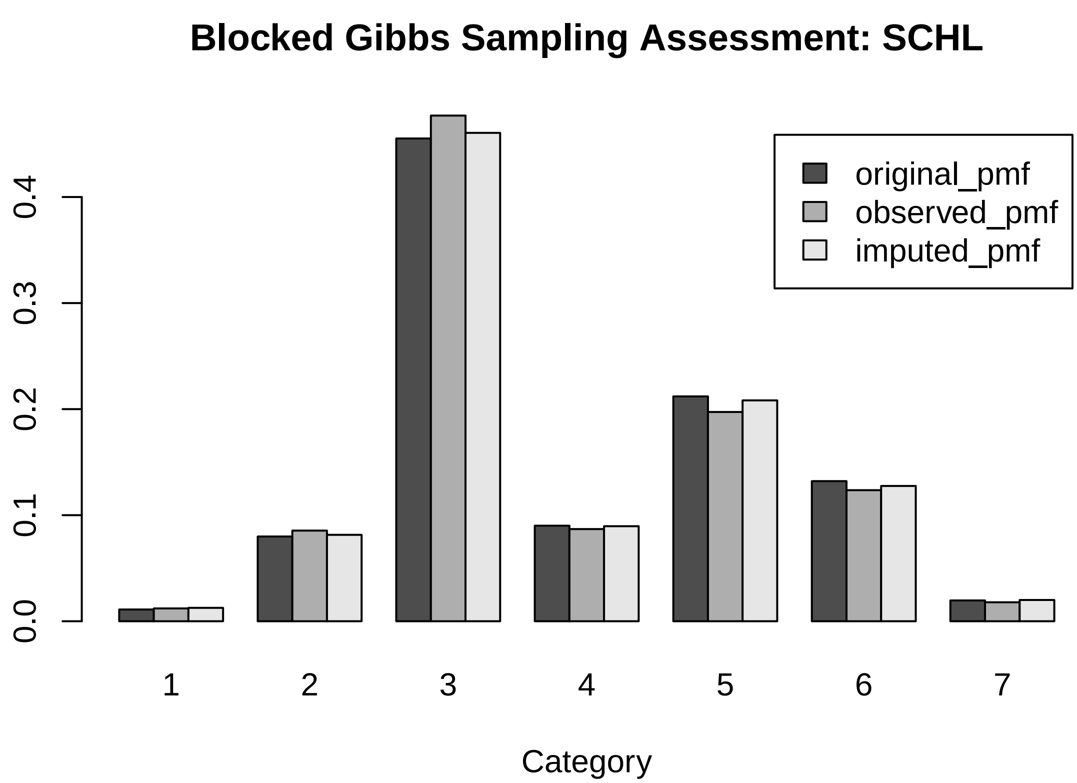
Diagnostics

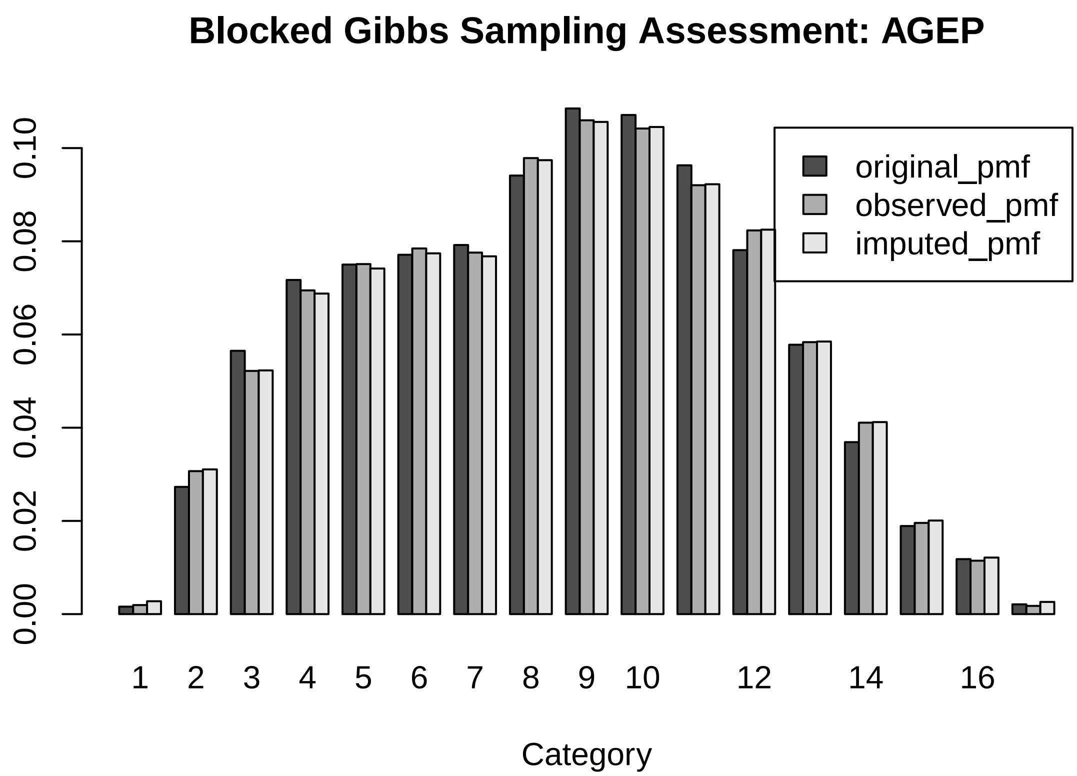


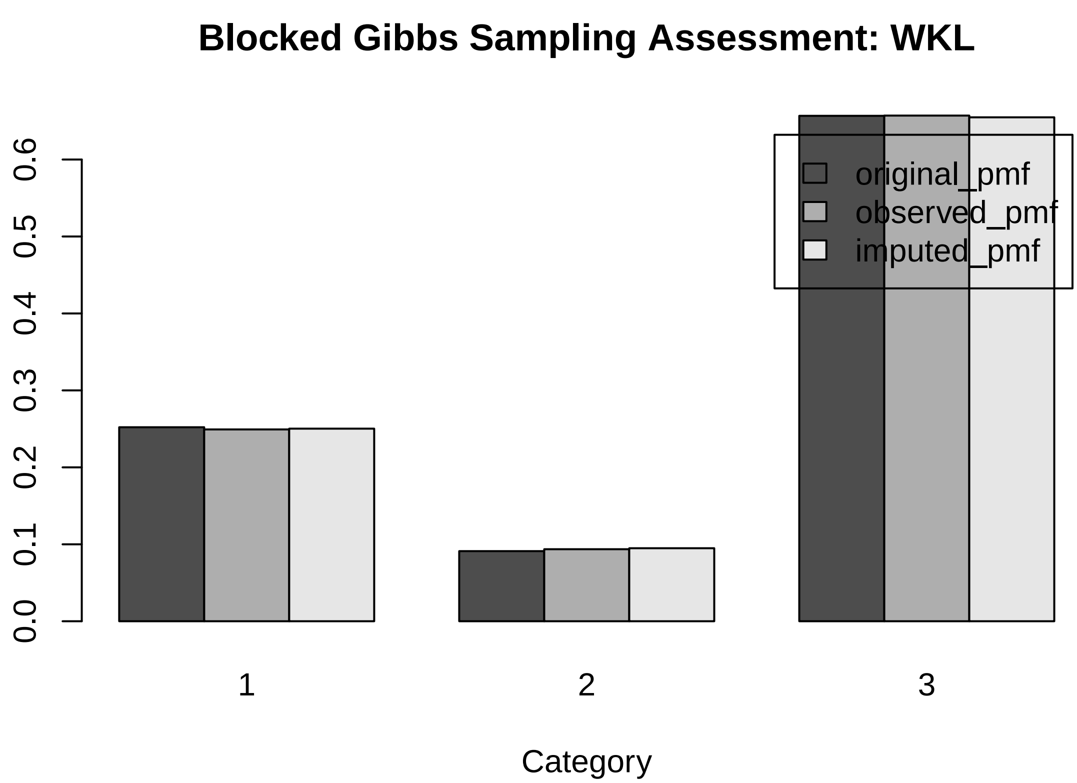
Assess bivariate joint distribution

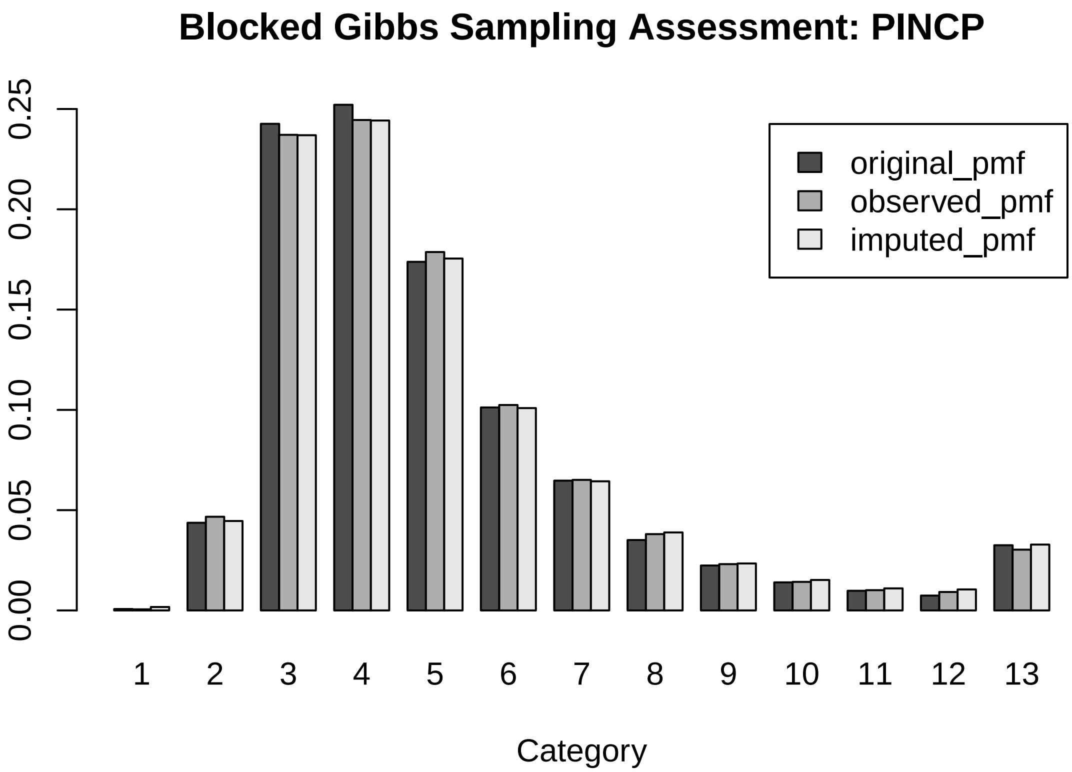
Assess trivariate joint distribution

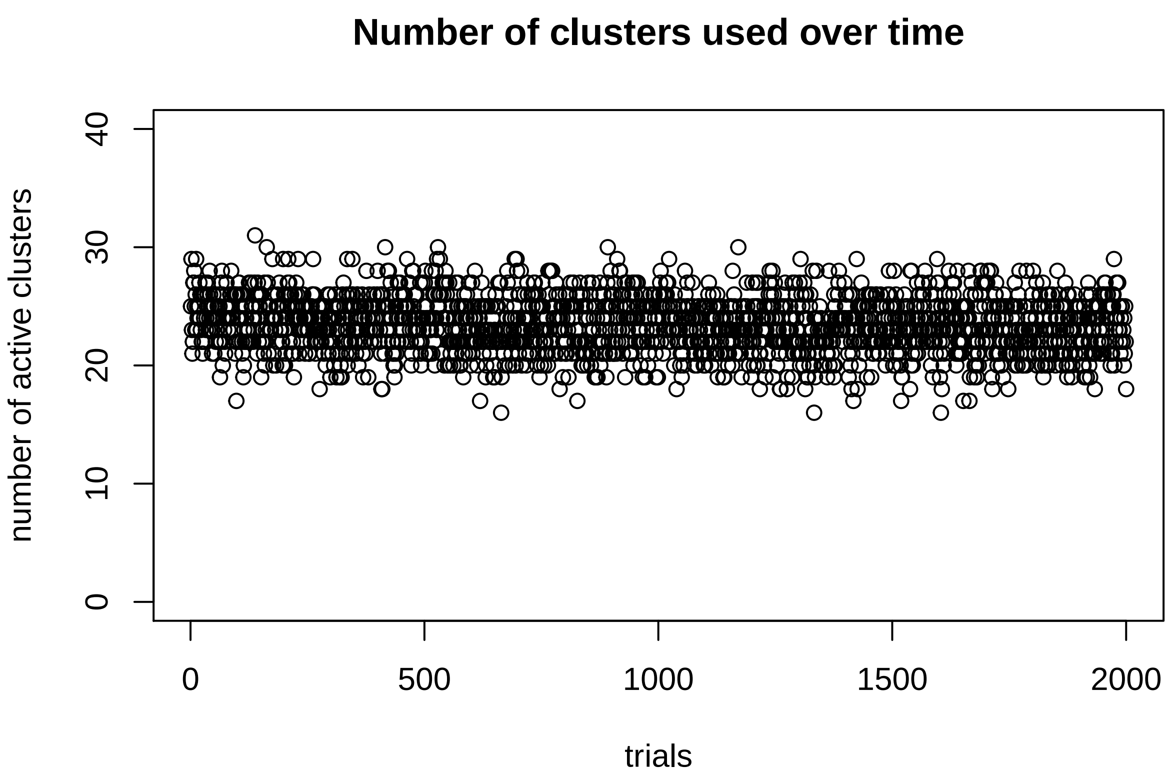

Blocked Gibbs Sampling Assessment: NP
original_pmf
observed_pmf
imputed_pmf
0.00
0.10
0.20
0.30
1
2
3
4
5
6
7
Category

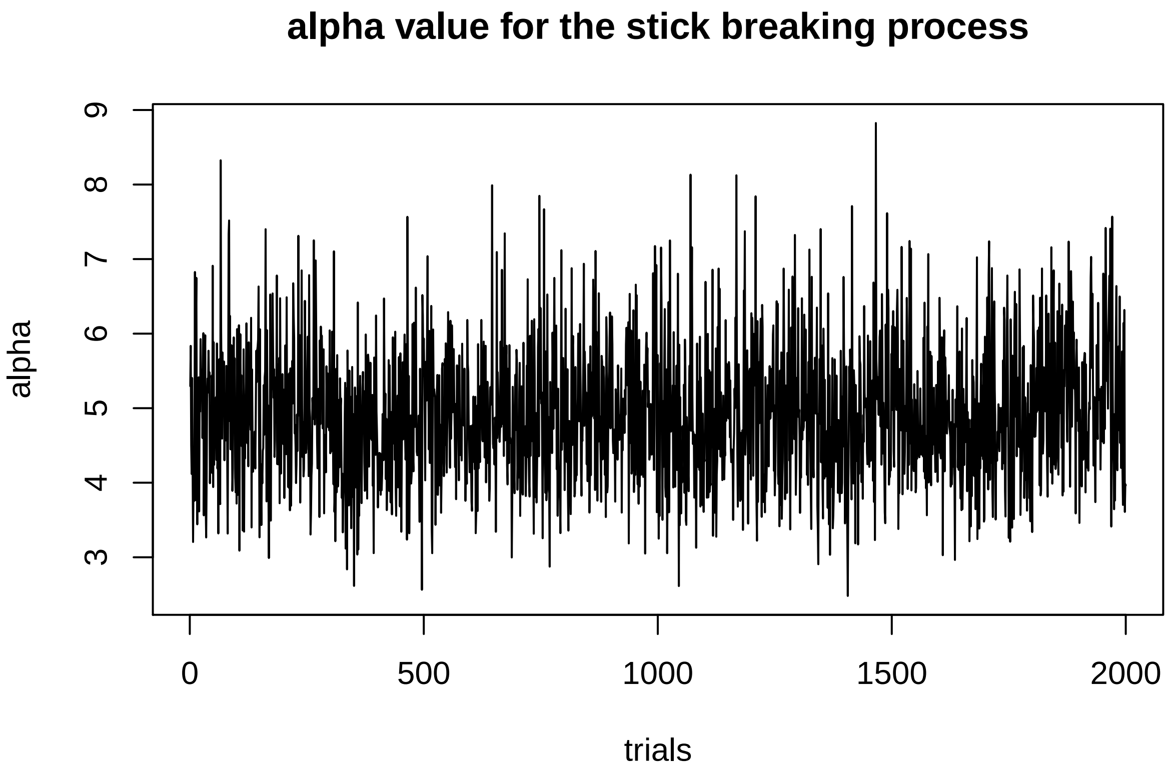

Blocked Gibbs Sampling Assessment: SCHL
original_pmf
observed_pmf
imputed_pmf
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
Category

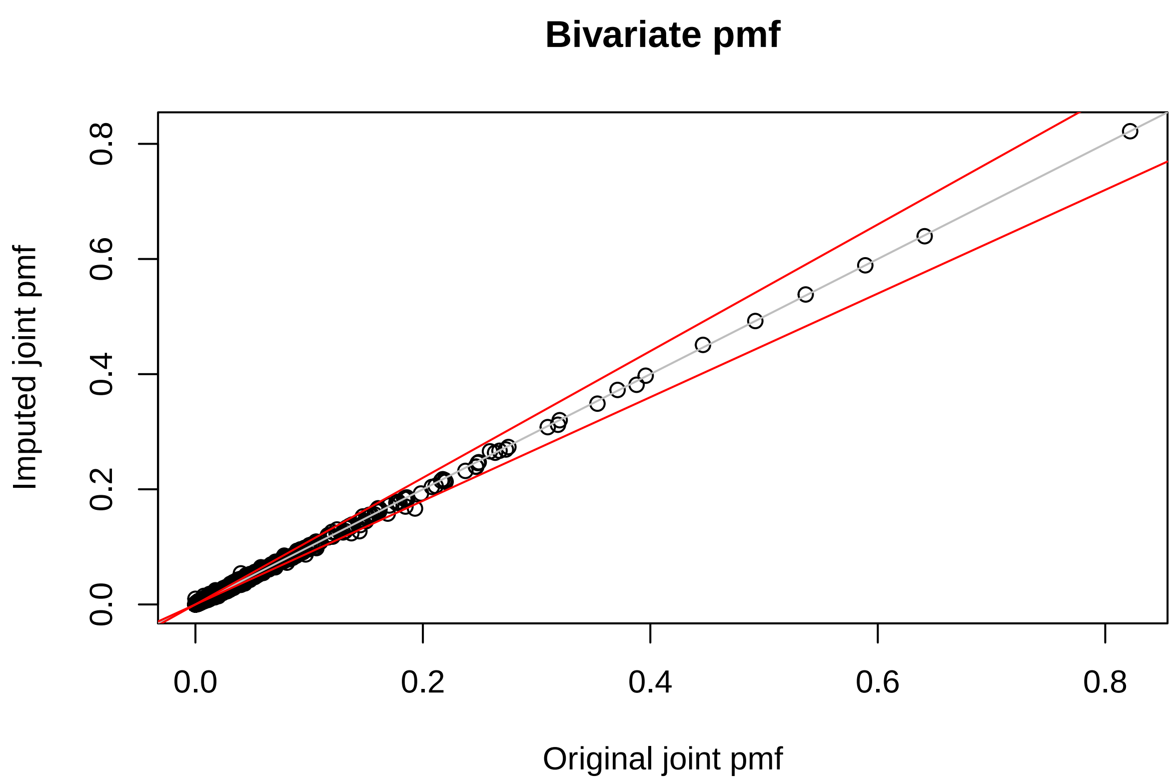

Blocked Gibbs Sampling Assessment: AGEP
0.00
0.02
0.04
0.06
0.08
0.10
original_pmf
observed_pmf
imputed_pmf
1
2
3
4
5
6
7
8
9
10
12
14
16
Category

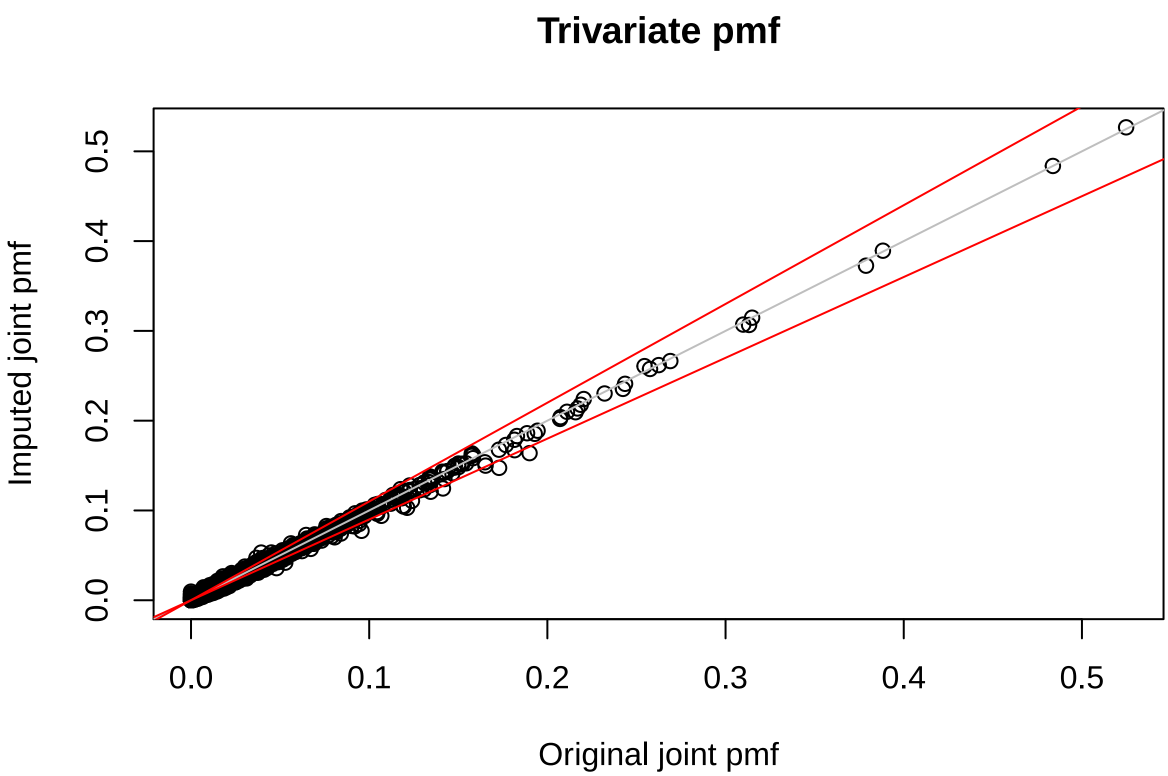

Blocked Gibbs Sampling Assessment: WKL
0.0
0.1
0.2
0.3
0.4
0.5
0.6
original_pmf
observed_pmf
imputed_pmf
1
2
3
Category


Blocked Gibbs Sampling Assessment: 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


Number of clusters used over time
number of active clusters
0
10
20
30
40
0
500
1000
1500
2000
trials


alpha value for the stick breaking process
alpha
trials
0
500
1000
1500
2000
3
4
5
6
7
8
9


Bivariate pmf
Imputed joint pmf
Original joint pmf
0.0
0.2
0.4
0.6
0.8



Trivariate pmf
Imputed joint pmf
Original joint pmf
0.0
0.1
0.2
0.3
0.4
0.5