Testing different imputation methods on PUMS (MCAR) - DPMPM

```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)
for (col in missing_col) {
  missing_ind <- rbernoulli(n,p = missing_prob)
  df_observed[missing_ind, col] <- NA
}
```

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 = 3000
- iter = 3000
- thinning = 5

4. Obtain result

```r
N = 40
Mon = 3000
B = 3000
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)

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

```
# 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)
```
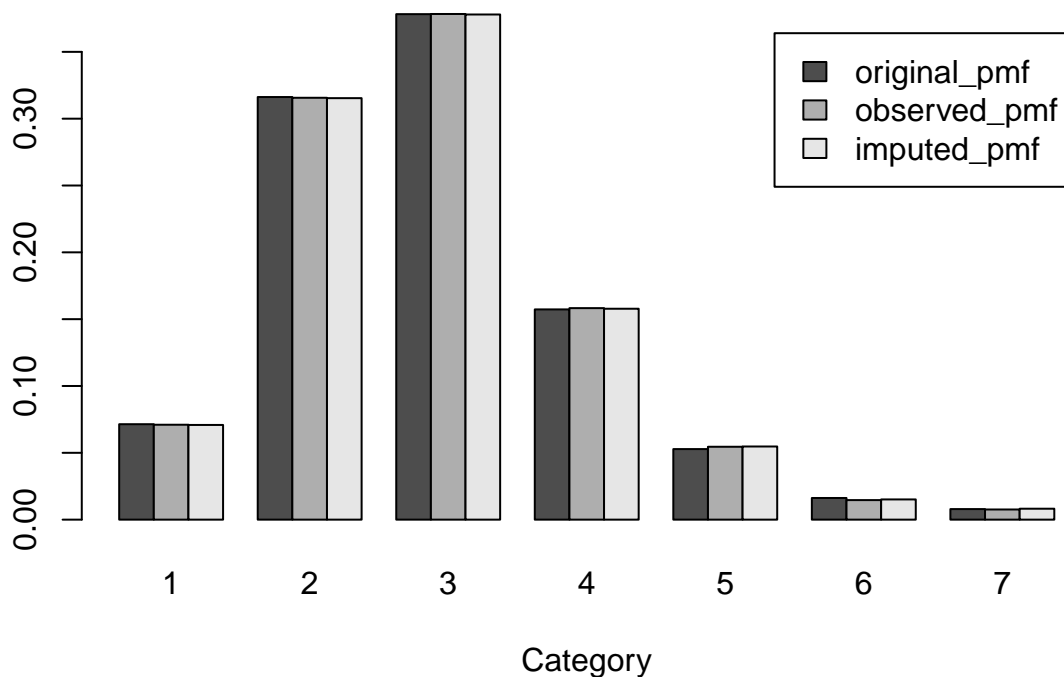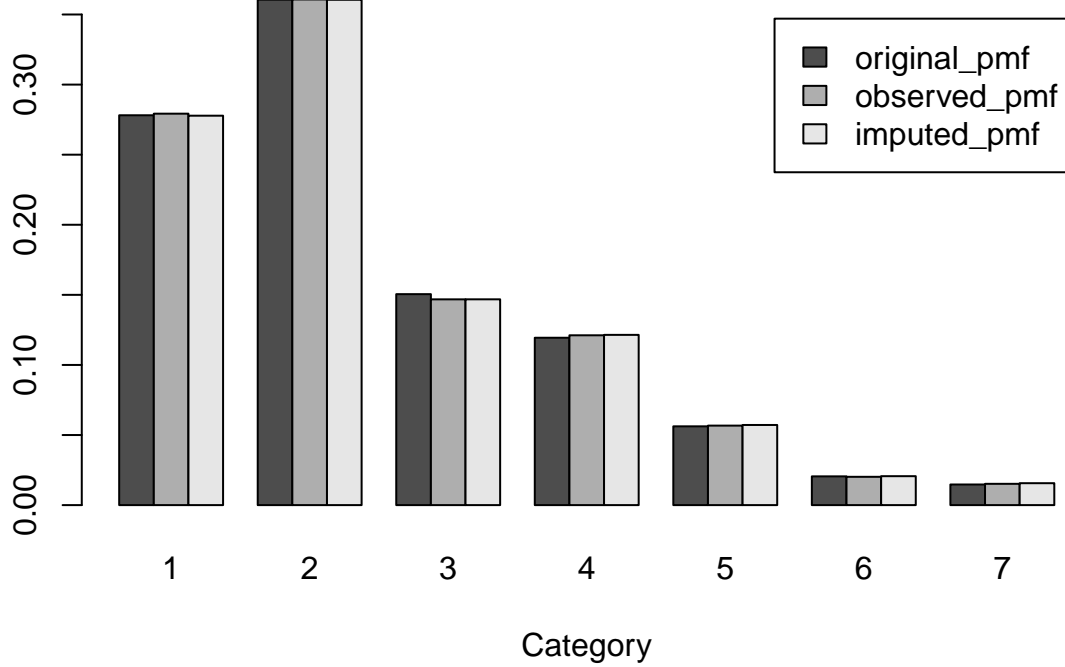
Diagnostics

Blocked Gibbs Sampling Assessment: VEH
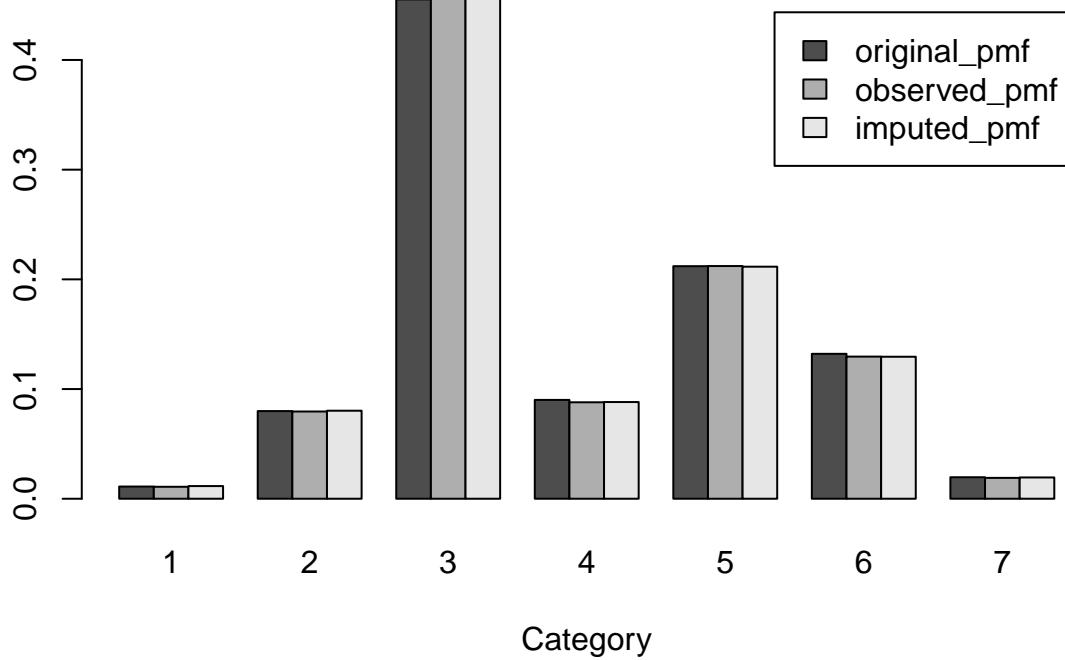

Assess bivariate joint distribution

Assess trivariate joint distribution

Blocked Gibbs Sampling Assessment: NP

Category

Blocked Gibbs Sampling Assessment: SCHL
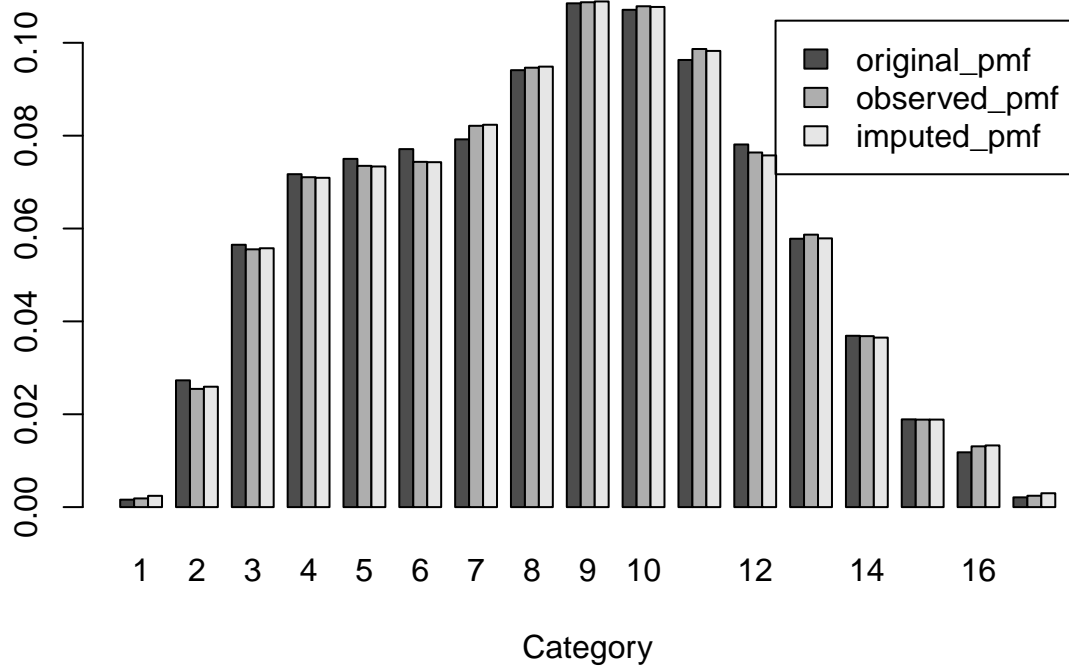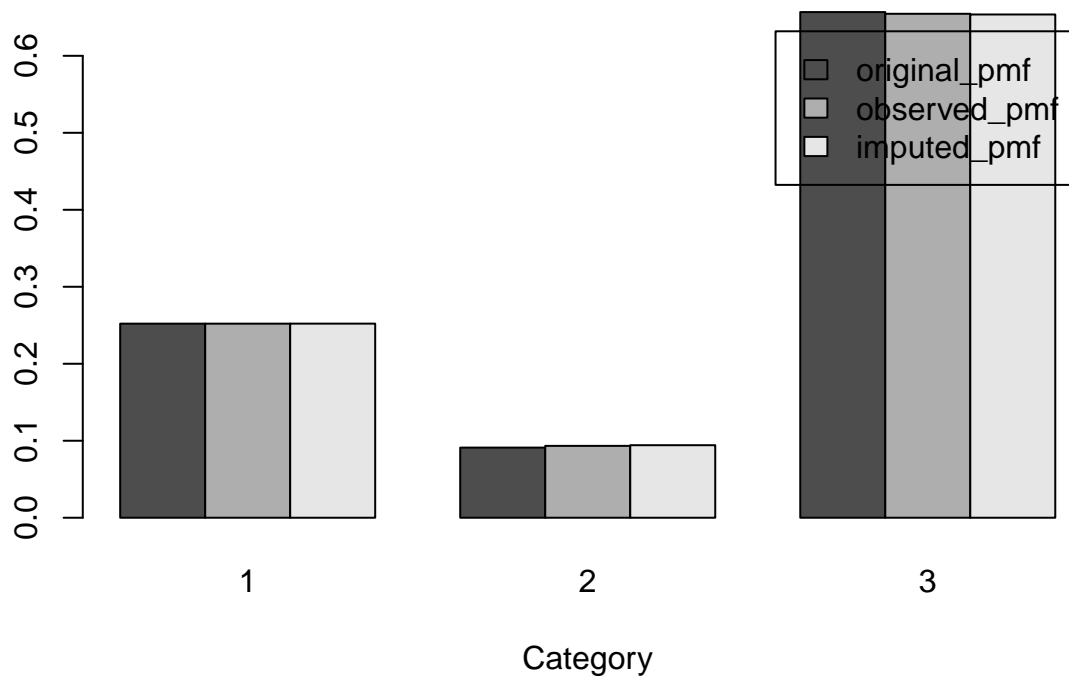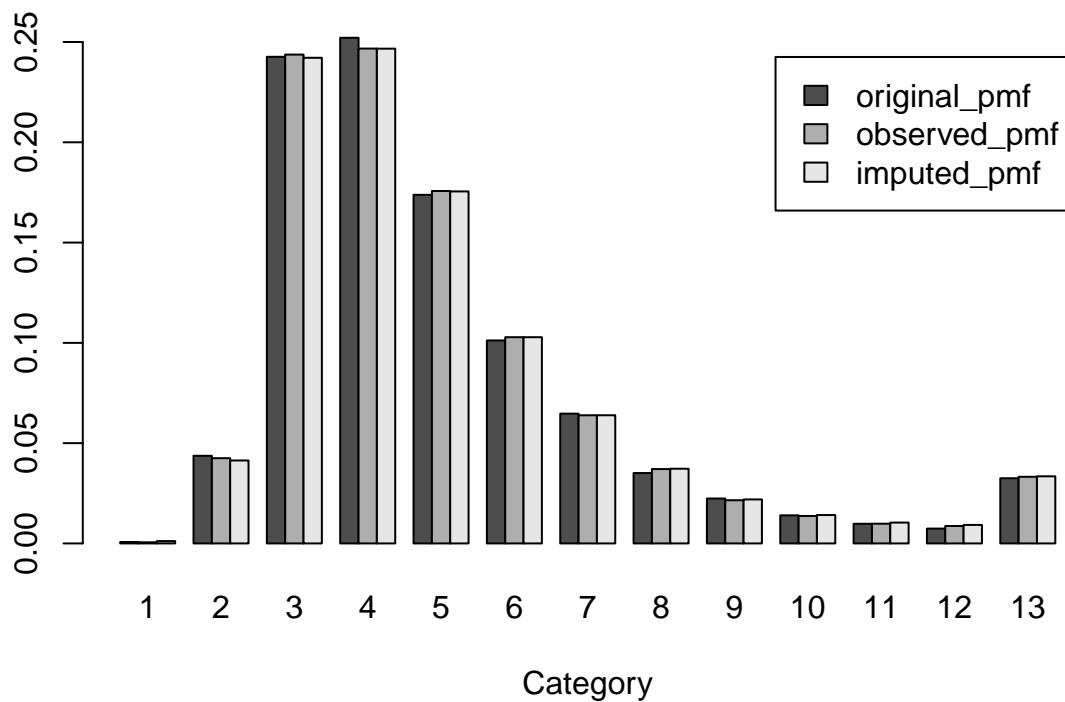

Category

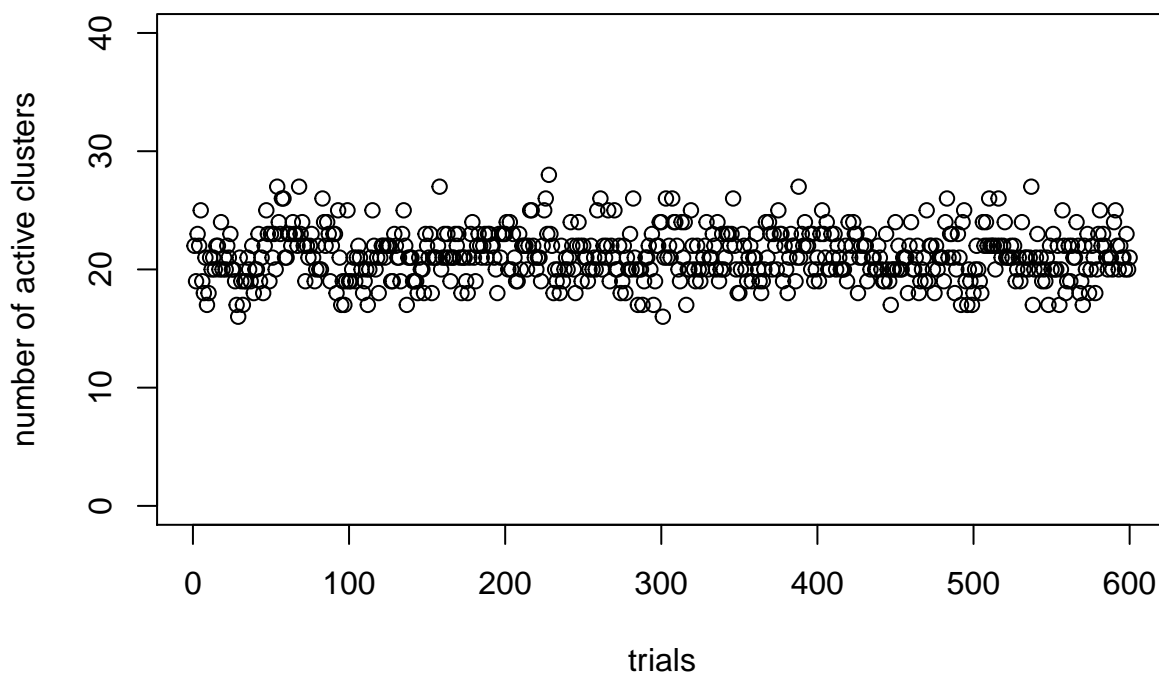# Blocked Gibbs Sampling Assessment: AGEP

Blocked Gibbs Sampling Assessment: WKL

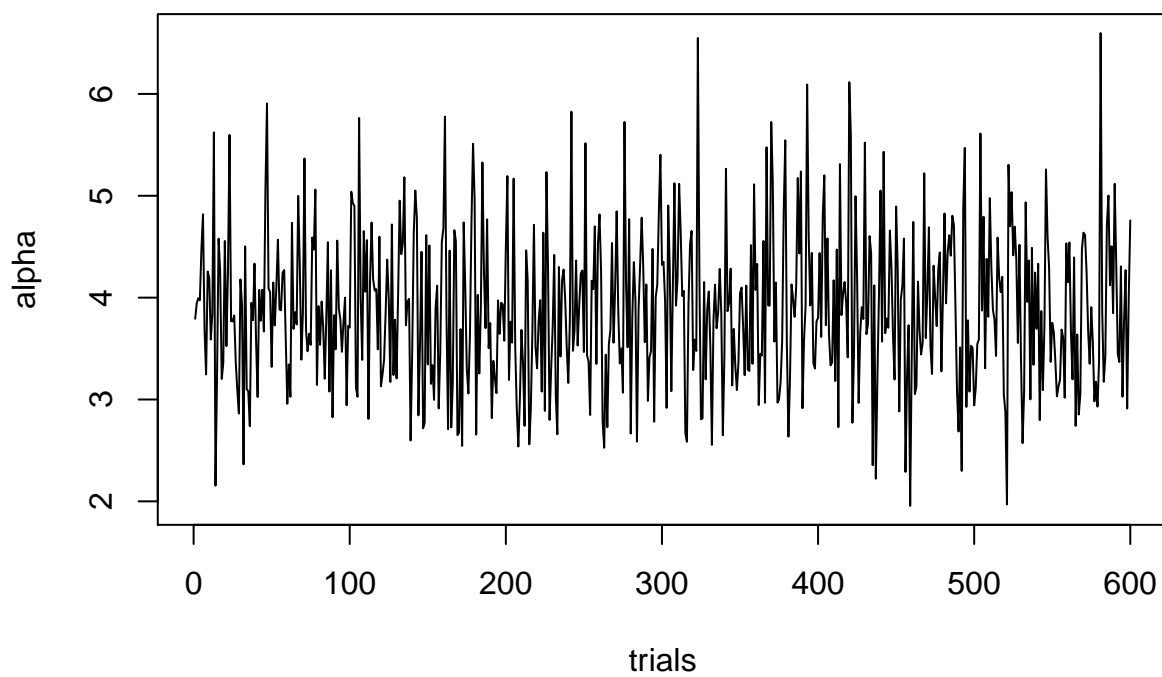## Blocked Gibbs Sampling Assessment: PINCP

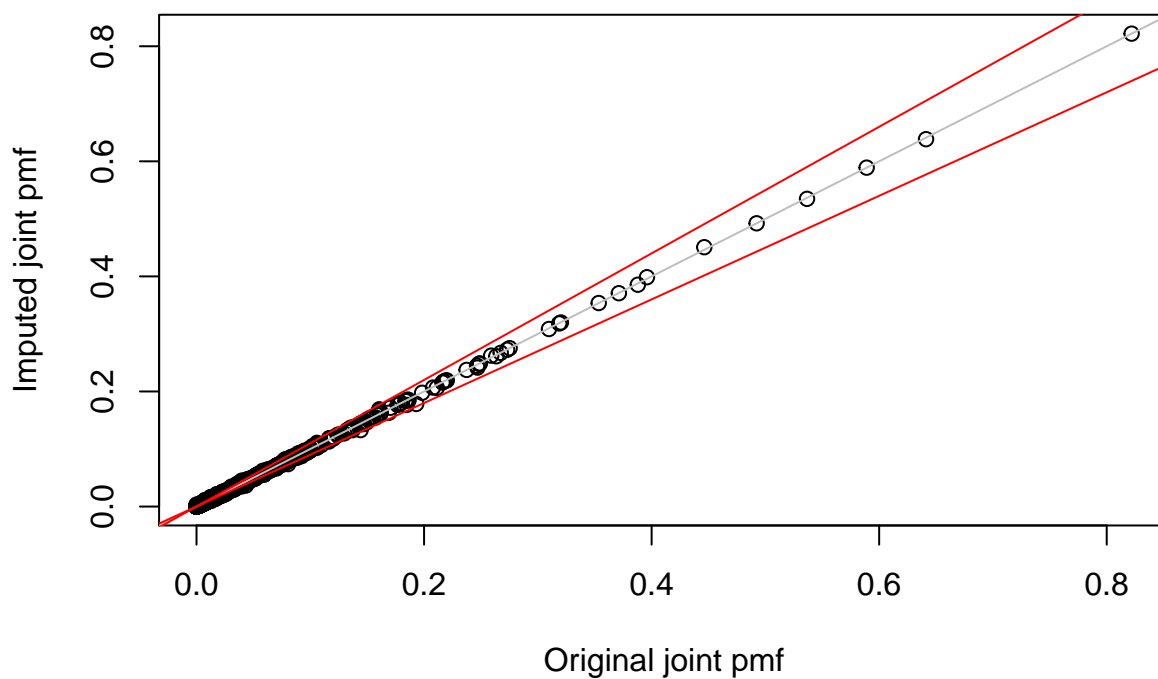## Number of clusters used over time

alpha value for the stick breaking process

trials

Bivariate pmf

Original joint pmf

Trivariate pmf