

# Testing different imputation methods on PUMS (MAR)

```
# 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 scenario 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)
beta_NP = c(-0.05, -1.5, 0.6, -2, -0.05)
beta0_NP = -0.05
beta_SCHL = c(-3, 3, -0.75, 0.05, -0.2)
beta0_SCHL = 0.05
beta_AGE = c(0.05, -0.2, 0.05, -1.25, 1)
beta0_AGE = -0.05
beta_PINCP = c(3, -0.05, -2.5, 0.05, -1)
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 AGE
prob_AGE = apply(t(t(normalized_df[, fully_observed_col])*beta_AGE)+beta0_AGE, 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

# 30.58% missing
apply(is.na(df_observed), MARGIN = 2, mean)

```

```

##      VEH      MV      NP      RMSP      ENG      MARHT      SCHL      RACNUM      AGEP      WKL      PINCP
## 0.3030 0.0000 0.3121 0.0000 0.0000 0.0000 0.2814 0.0000 0.3355 0.3017 0.3011

```

### Ordinal bayesian nonparametric model

```

source("../..//probitBayes.R")
N = 40
Mon = 300
B = 300
thin.int = 1
# function(y, N = 40, Mon = 2000, B = 300, thin.int = 5, seed = 0)
output_list <- probitBayesImputation(df_observed, N, Mon, B, thin.int)

sampled_y <- output_list[['sampled_y']]
sampled_z <- output_list[['sampled_z']]

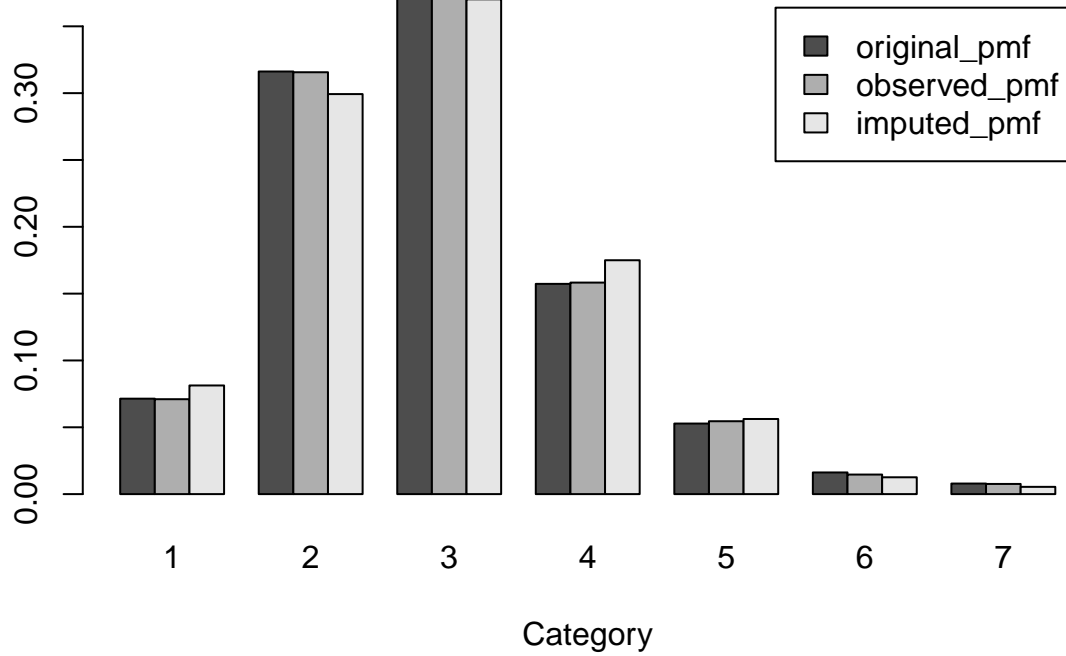
```

Diagnostics

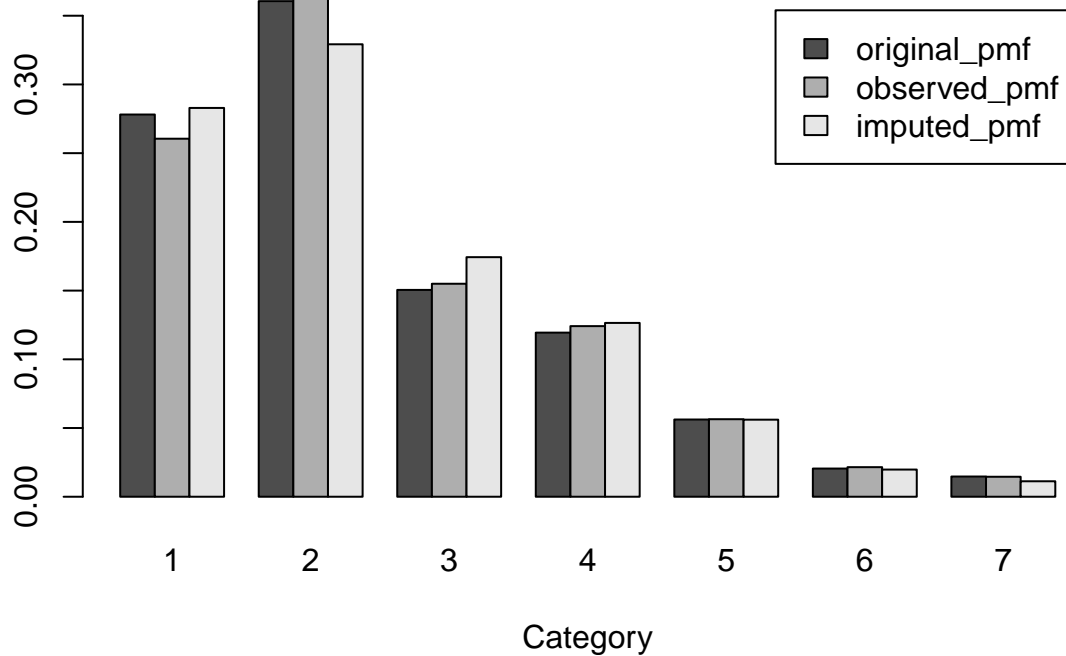
Assess bivariate joint distribution

Assess trivariate joint distribution

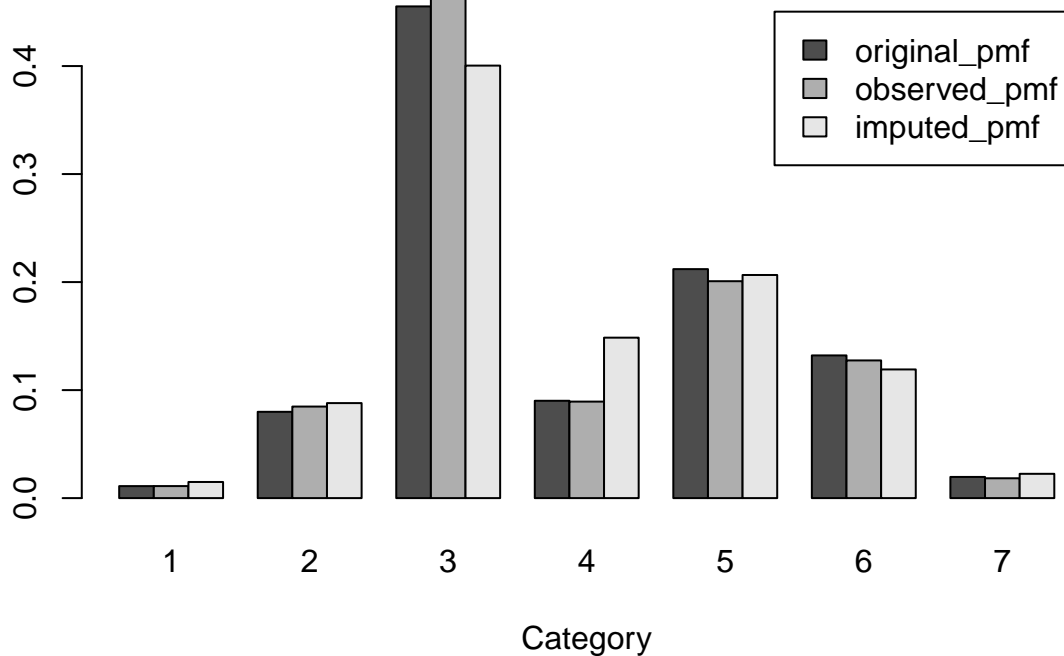
### Blocked Gibbs Sampling Assessment: VEH



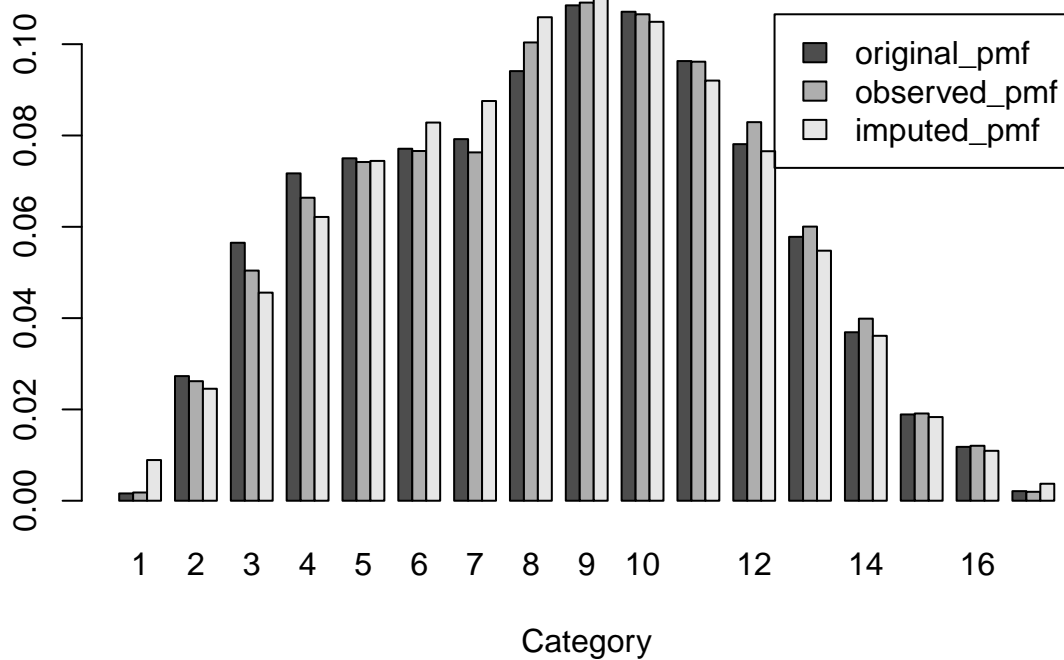
### Blocked Gibbs Sampling Assessment: NP



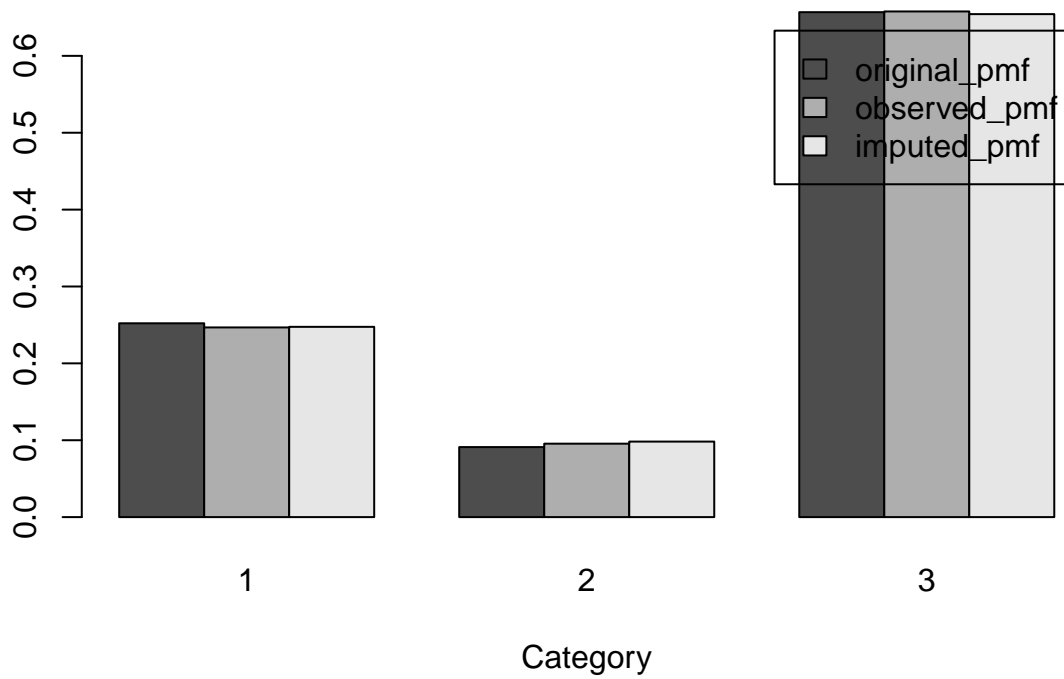
### Blocked Gibbs Sampling Assessment: SCHL



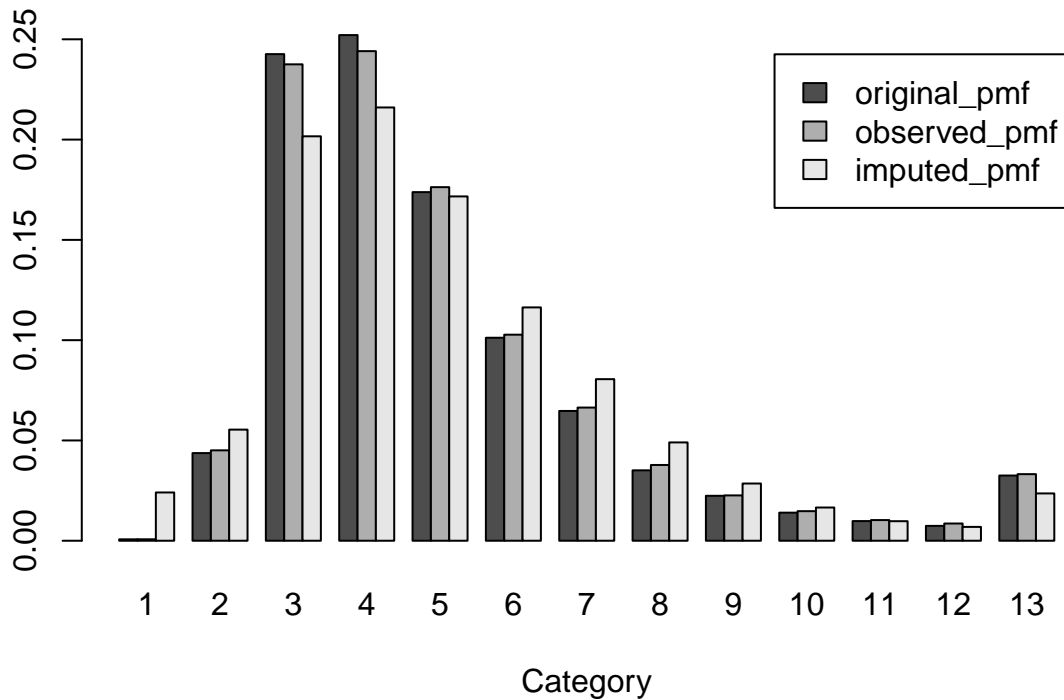
### Blocked Gibbs Sampling Assessment: AGEF



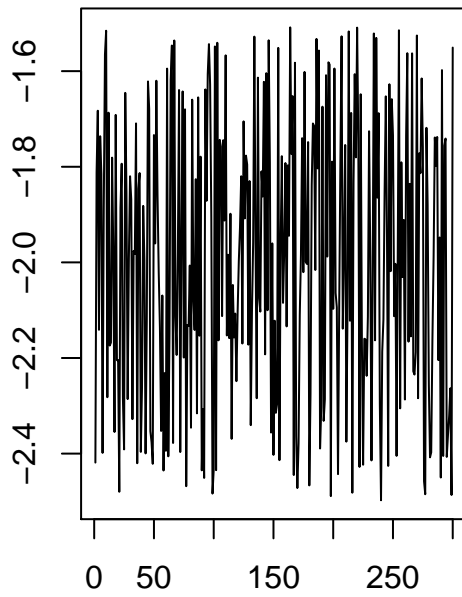
### Blocked Gibbs Sampling Assessment: WKL



### Blocked Gibbs Sampling Assessment: PINCP

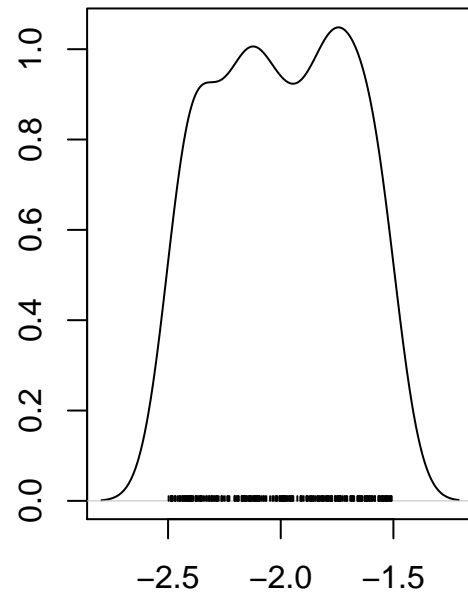


**Trace of var1**

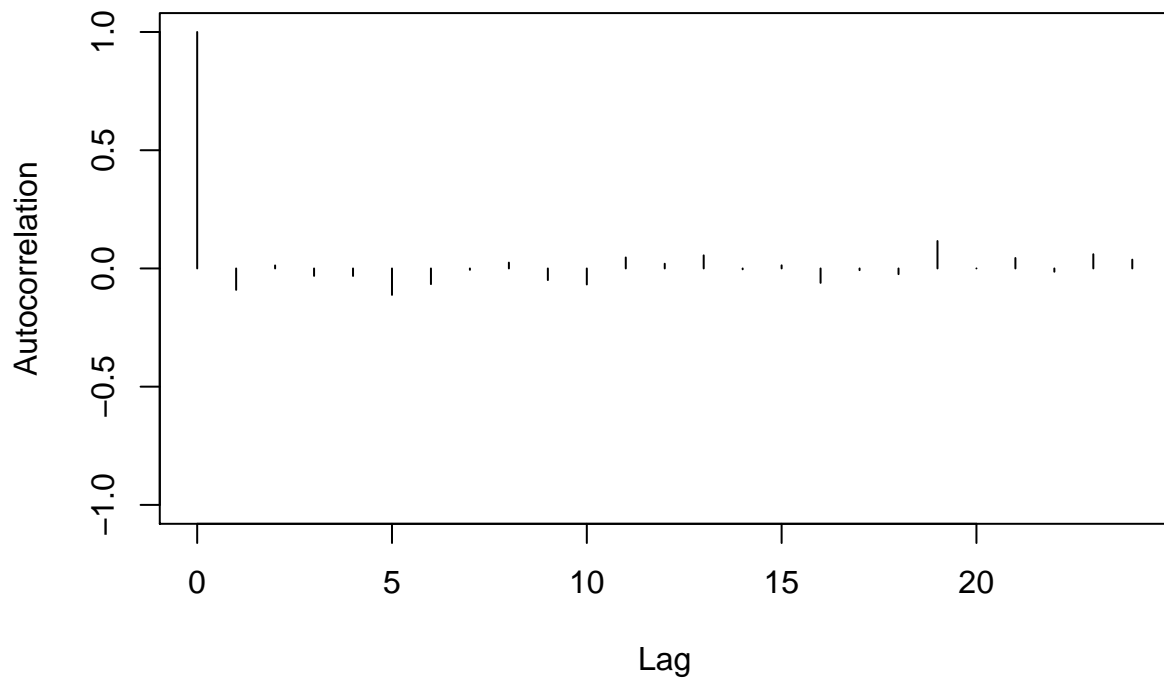


Iterations

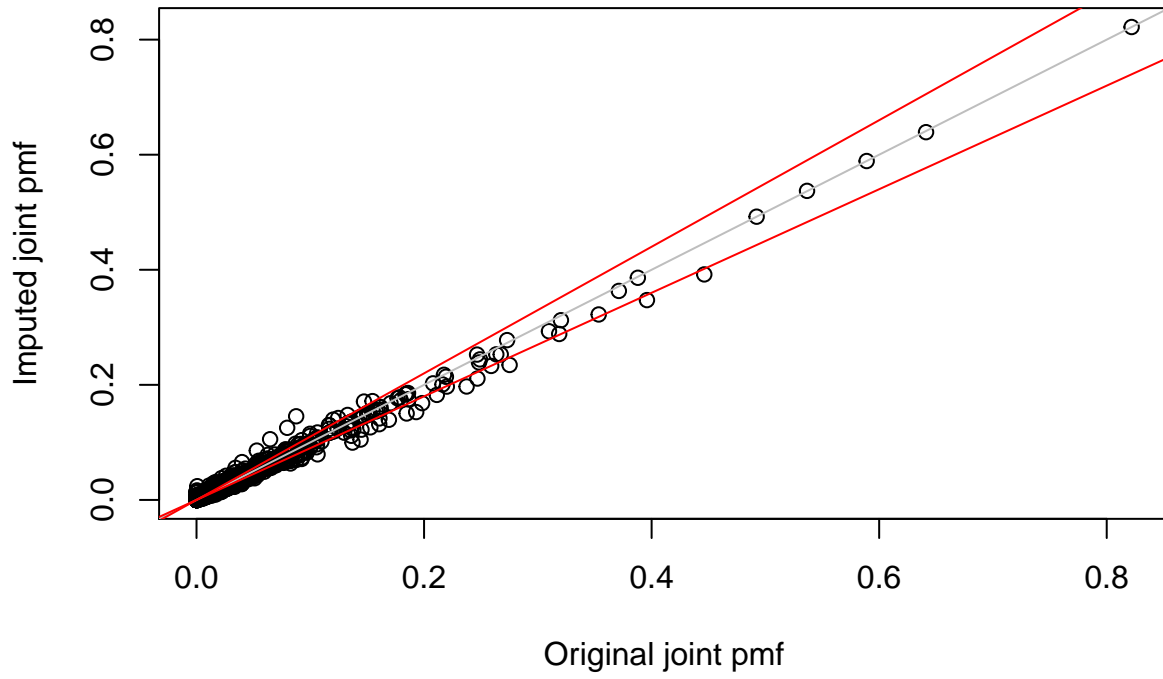
**Density of var1**



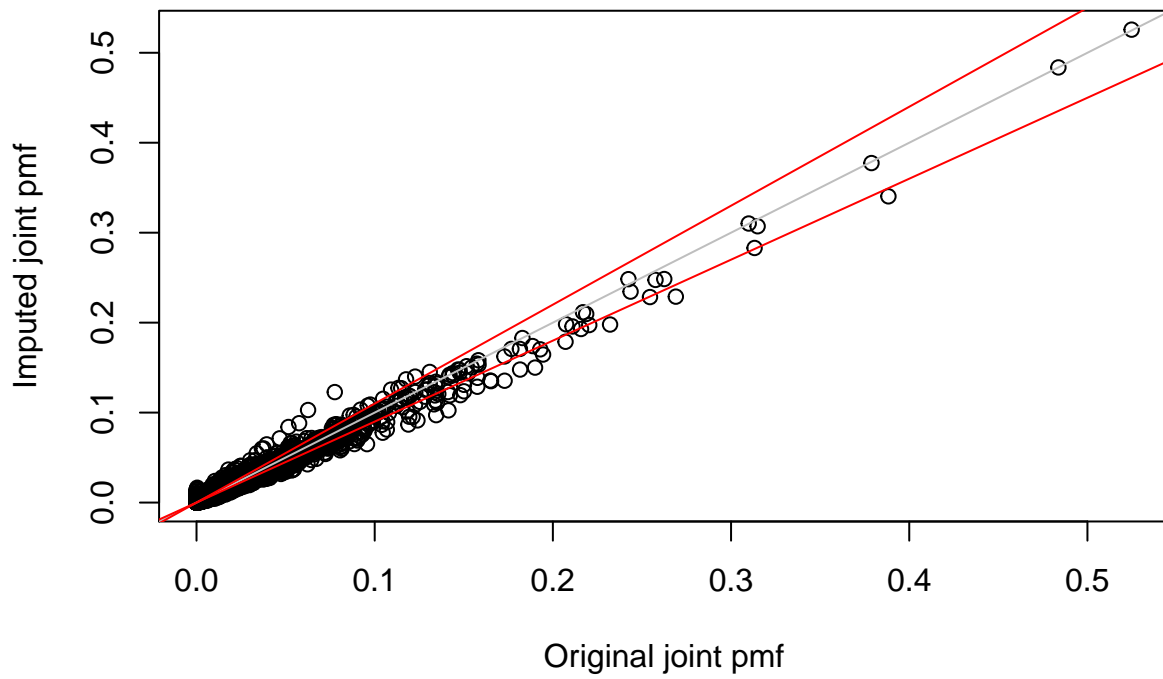
N = 300 Bandwidth = 0.09944



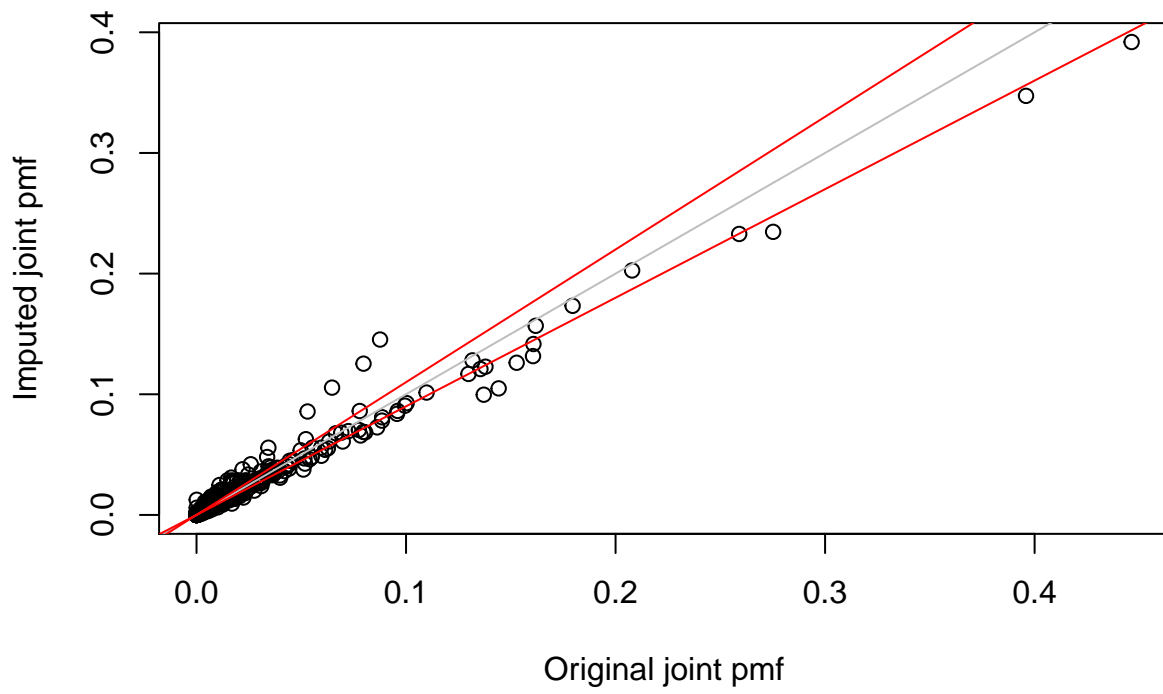
**Bivariate pmf**



**Trivariate pmf**



**Bivariate pmf SCHL**



**Bivariate pmf WKL**

