

Testing different imputation methods on PUMS (MAR)

- MICE

```
# 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_AGE = c(0.05, -0.2, 0.05, -1.25, 1)+c(0,0,0,1.1,1.1)
beta0_AGE = -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

```

MICE

Create 5 imputed dataset

```

library(mice)

##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
##      cbind, rbind

imputed_df <- mice(df_observed,m=5,print=F)

```

```
## Warning: Number of logged events: 150
```

Extract the 5 imputed dataset

```

d1 <- complete(imputed_df, 1)
d2 <- complete(imputed_df, 2)
d3 <- complete(imputed_df, 3)
d4 <- complete(imputed_df, 4)
d5 <- complete(imputed_df, 5)
imputed_sets = rbind(d1, d2, d3, d4, d5)

```

Diagnostics

Assess bivariate joint distribution

Assess trivariate joint distribution

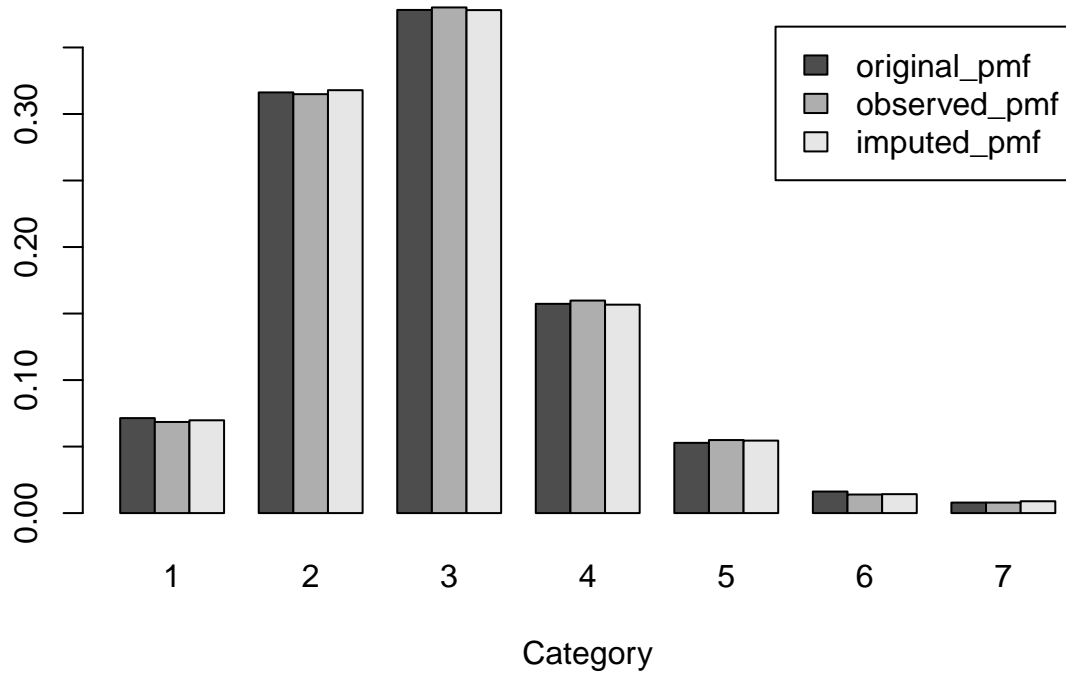
```

# 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)

```

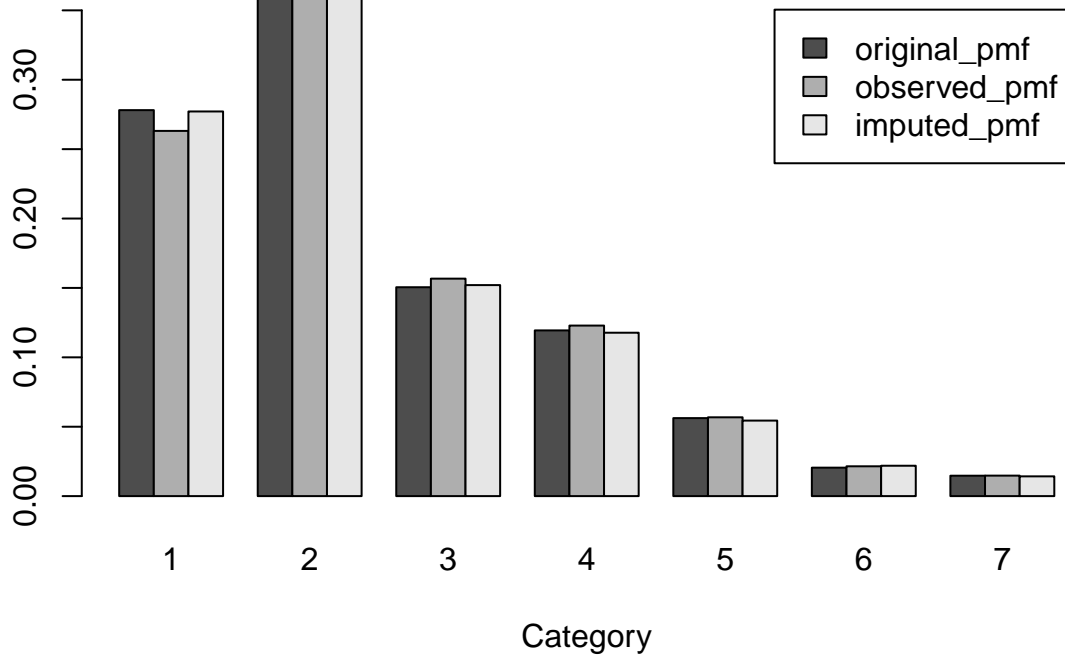
MICE: VEH



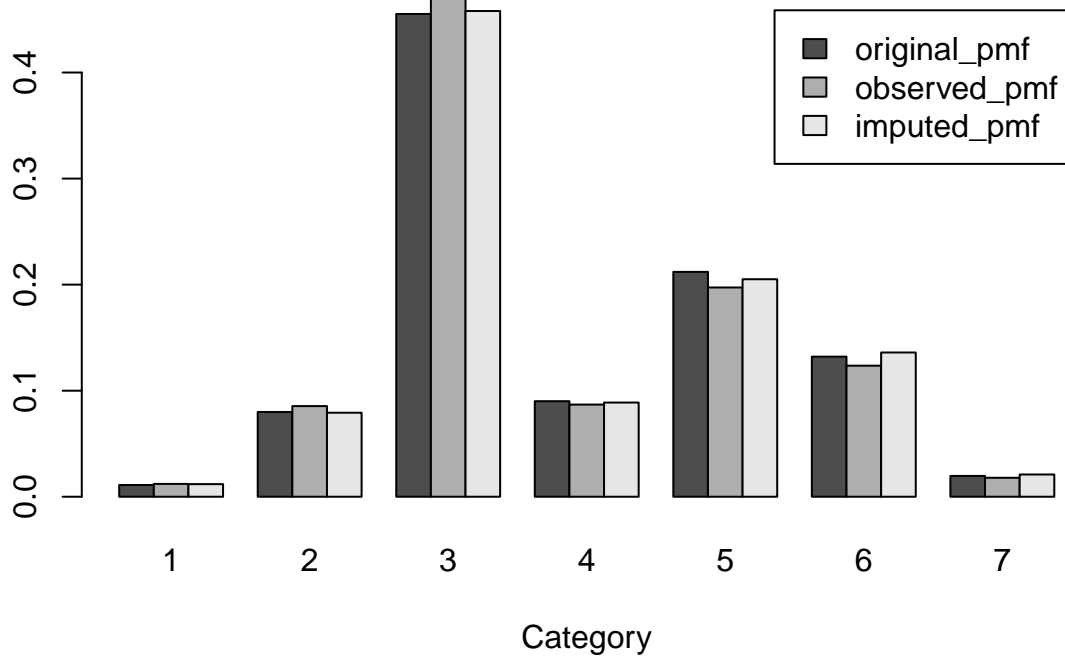
```
rmse = sqrt(sum((normalized_df[missing_matrix] - normalized_impute[missing_matrix])^2)/sum(missing_matrix))
rmse
```

```
## [1] 0.3183811
```

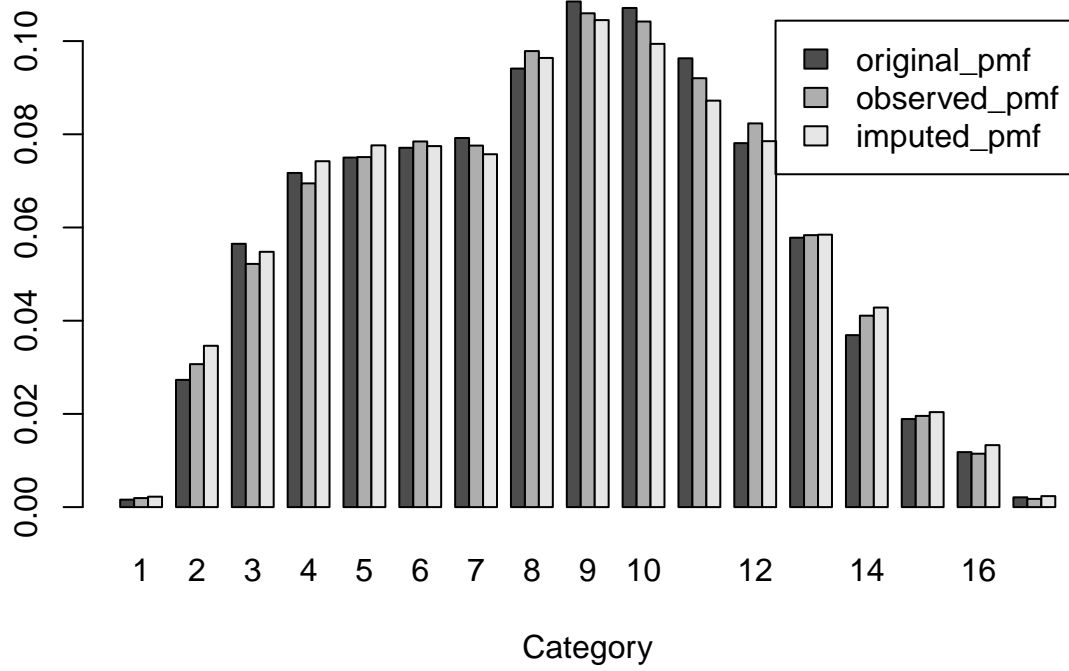
MICE: NP



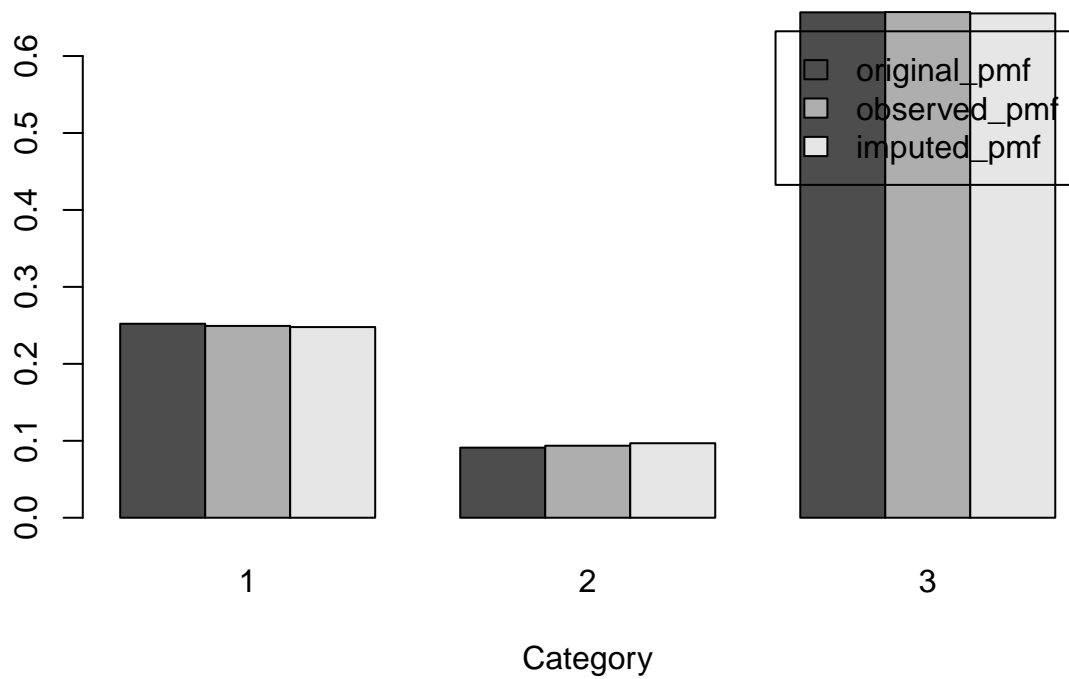
MICE: SCHL



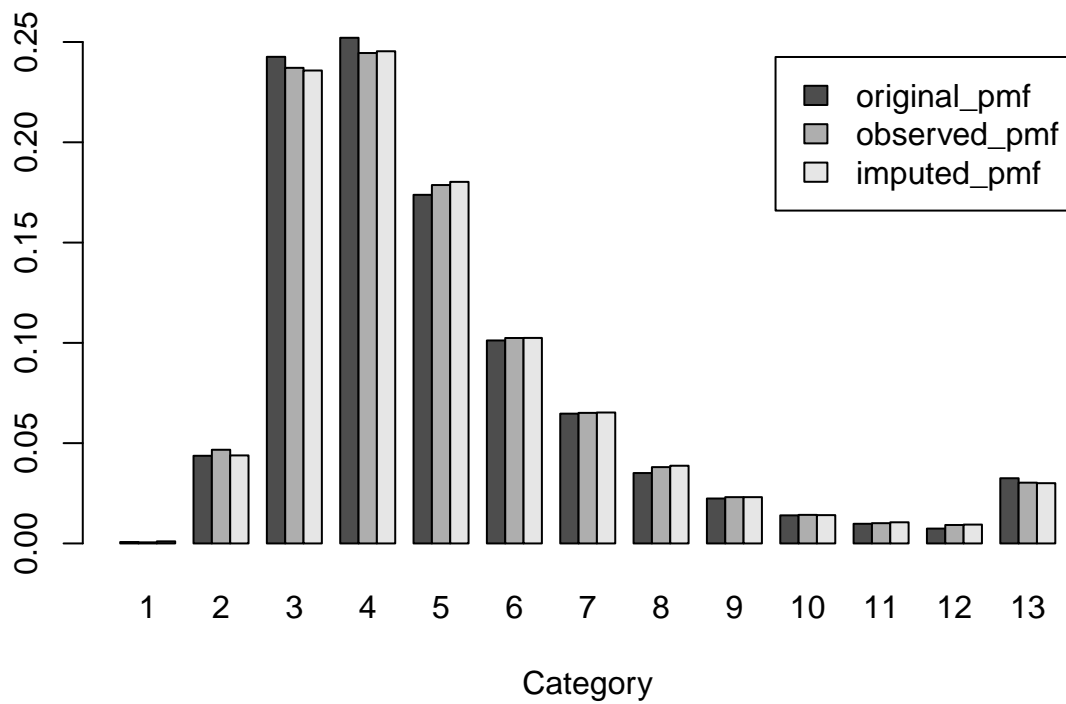
MICE: AGEP



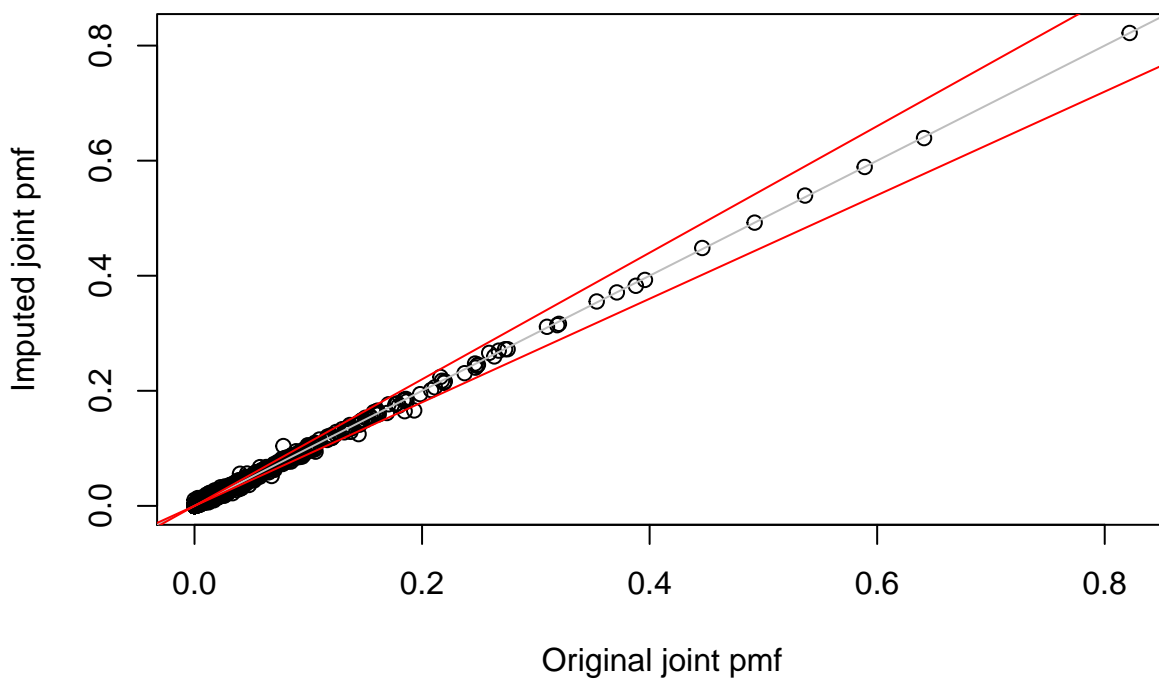
MICE: WKL



MICE: PINCP



Bivariate pmf



Trivariate pmf

