

Testing different imputation methods on PUMS (MCAR) - 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 scenario 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
}
```

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)
```

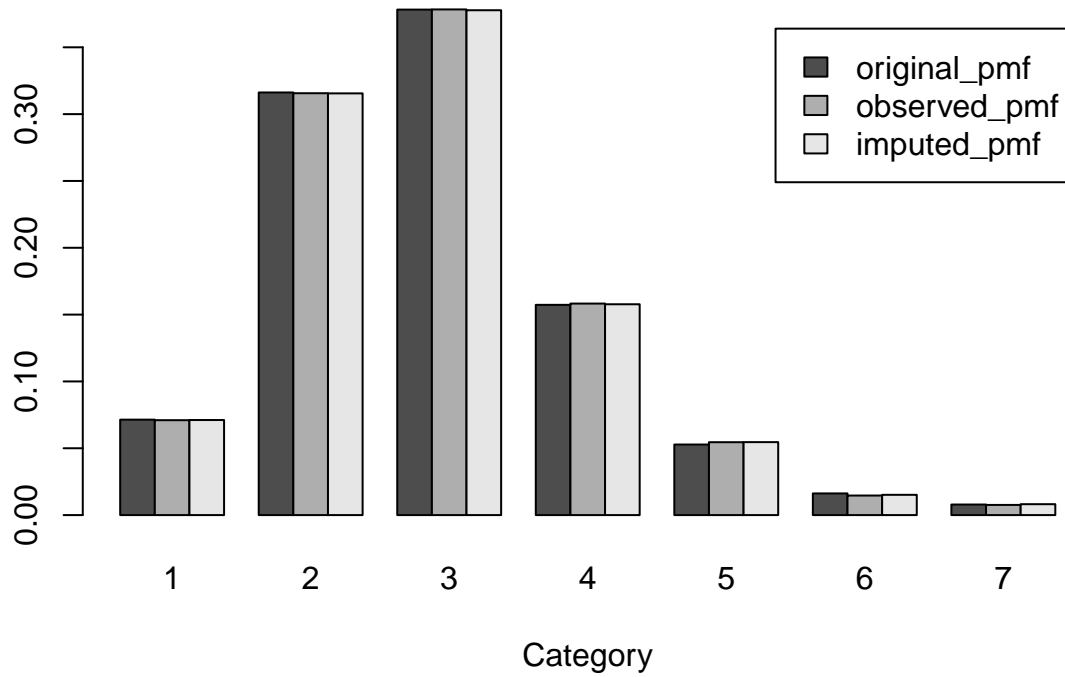
Diagnostics

Assess bivariate joint distribution

Assess trivariate joint distribution

```
# calculate rmse
numeric_df = sapply(df, as.numeric)
```

MICE: VEH



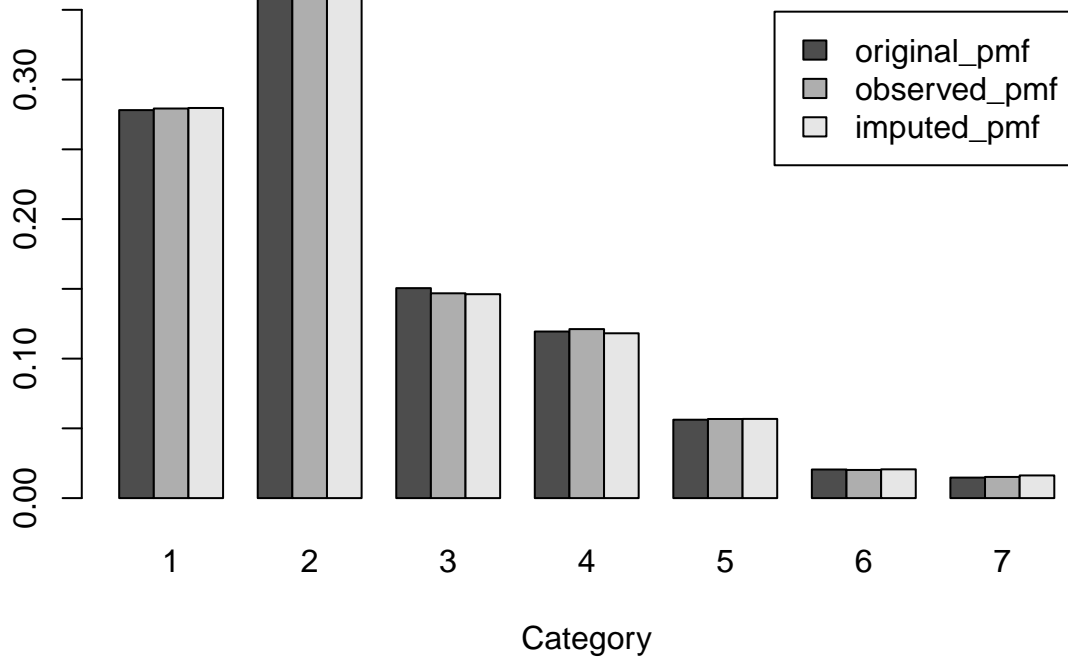
```
normalized_df = t(t(numeric_df-1)/(apply(numeric_df, MARGIN = 2, FUN = max)-1))
numeric_impute = sapply(d1, as.numeric)
normalized_impute = t(t(numeric_impute-1)/(apply(numeric_df, MARGIN = 2, FUN = max)-1))

missing_matrix = is.na(df_observed)

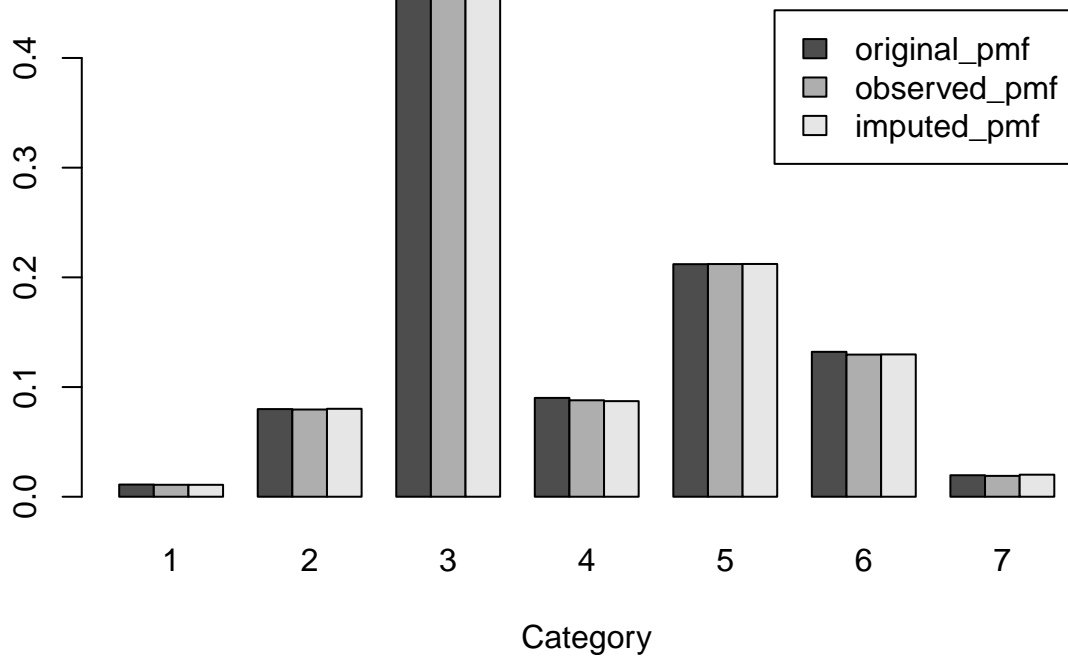
rmse = sqrt(sum((normalized_df[missing_matrix] - normalized_impute[missing_matrix])^2)/sum(missing_matrix))
rmse

## [1] 0.3145641
```

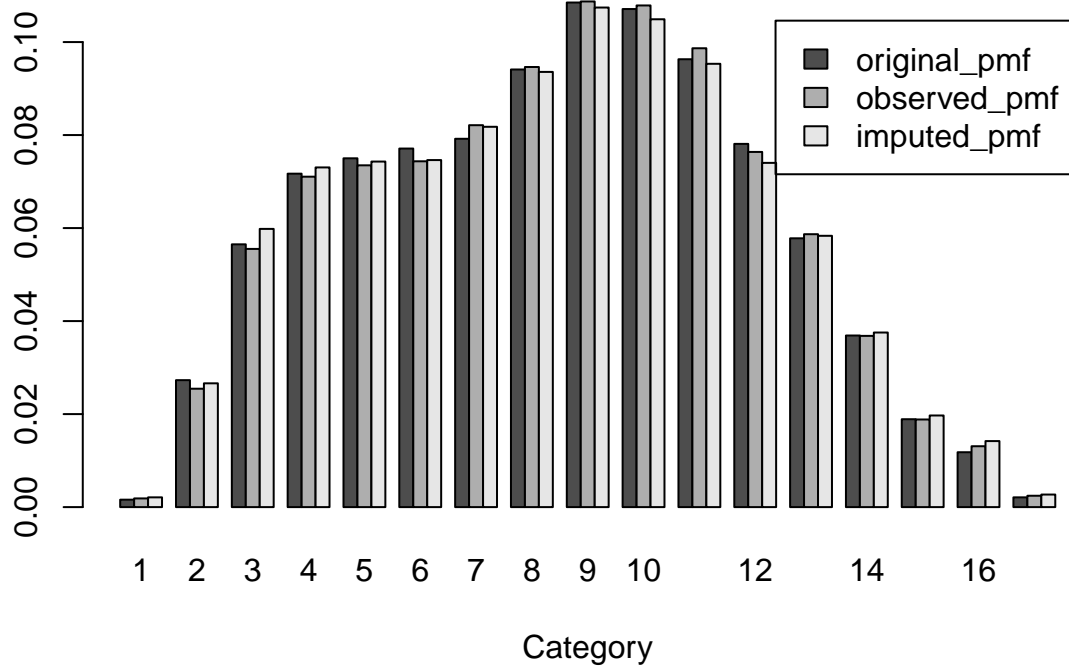
MICE: NP



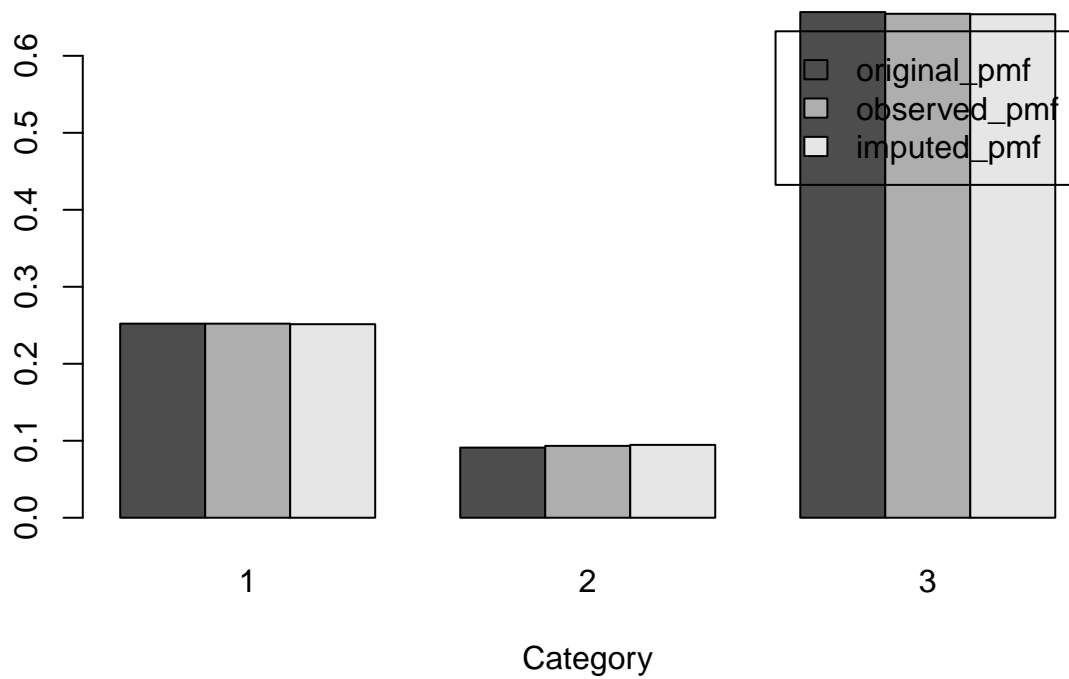
MICE: SCHL



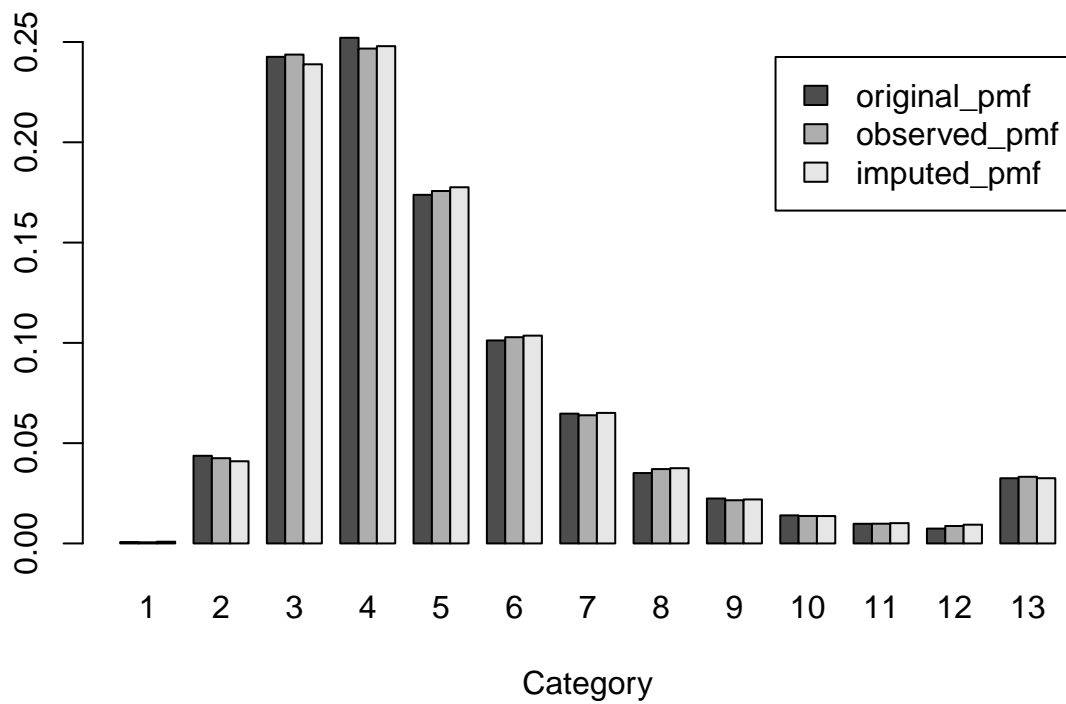
MICE: AGEP



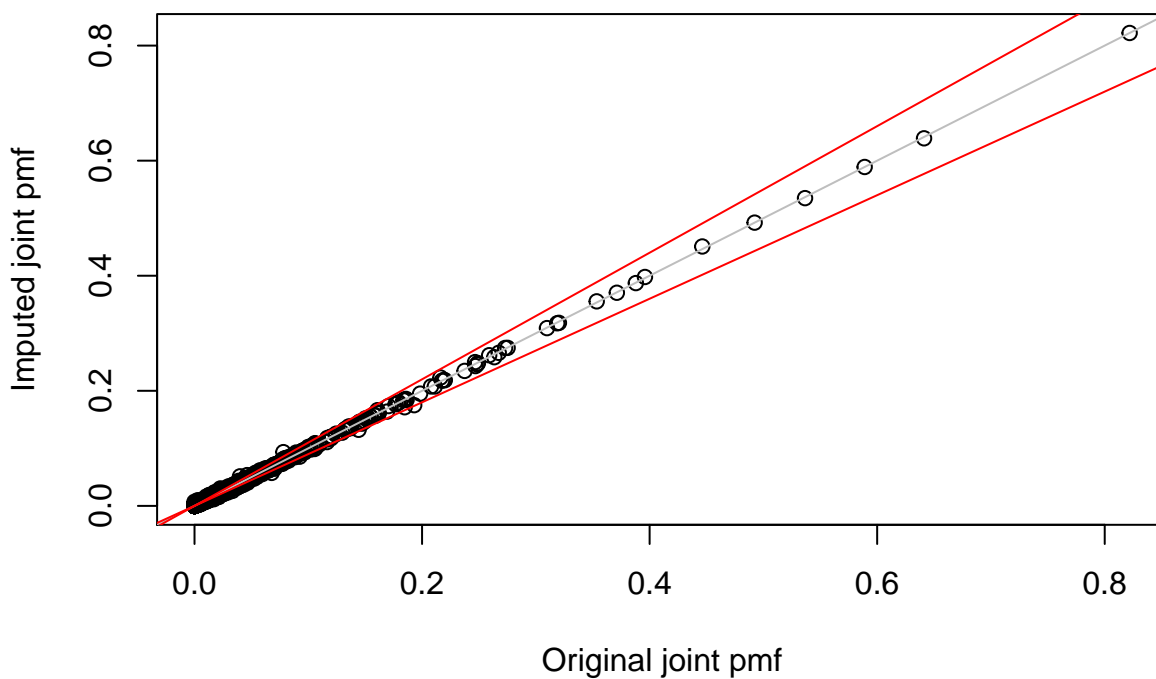
MICE: WKL



MICE: PINCP



Bivariate pmf



Trivariate pmf

