Missing Data Imputaion on HR Data Set

Juntao Zhang

13/03/2021

##Objective The goal of this project is to impute missing data using different methods for the HR data set from Kaggle. For the purpose of the project, at least one continuous variable and one categorical variable with a minimum of 20% missingness are required.

####Introduction

A company is providing training for future employees and they have a data set of the people who registered for the training. There are 19158 observations and 11 variables in the data set.

Variables (explanation, % missing and type):
enrollee_id: Unique ID for candidate, 0 NA, continuous variable
City: City code, 0 NA, continuous variable
Gender: Gender of candidate, about 23% NAs
Relevent_experience: relevant experience of candidate, no NA
Enrolled_university: Type of University course enrolled if any, about 2% NAs, categorical variable
Education_level: Education level of the candidate, about 2.3% NAs, categorical variable
Experience: Candidate total experience in years, no NA, continuous variable
Company_size: Number of employees in current employer's company, about 31% NAs, categorical variable
Lastnewjob: Difference in years between previous job and current job, about 2% NAs, categorical variable
Training_hours:Training hours completed, 0 NAs, continuous variable
Target: 0-Not looking for job change, 1-looking for job change, 0 NAs, binary variable

After data imputation, an analysis on how to identify which of the candidates really wants to work for the company (i.e looking for a job change) will be performed. A logistic regression model will be fitted to the data with the target variable (binary with values 0 and 1) as the outcome variable and the other variables as the covariates. The analysis can find out the effects of the covariates on the identifier variable: target.

```r
data <-read.csv("C:/Users/joann/OneDrive/Desktop/missing data/week 2/aug_train.csv",
                 na.strings = "")
#Inspect dataset
str(data)
```

```
## 'data.frame':    19158 obs. of  11 variables:
##  $ enrollee_id        : int  8949 29725 11561 33241 666 21651 28806 402 27107 699 ...
##  $ city               : chr  "city_103" "city_40" "city_21" "city_115" ...
##  $ gender             : chr  "Male" "Male" NA NA ...
##  $ relevent_experience: chr  "Has relevent experience" "No relevent experience" "No relevent experie
##  $ enrolled_university: chr  "no_enrollment" "no_enrollment" "Full time course" NA ...
##  $ education_level    : chr  "Graduate" "Graduate" "Graduate" "Graduate" ...
##  $ experience         : chr  ">20" "15" "5" "<1" ...
##  $ company_size       : chr  NA "50-99" NA NA ...
##  $ last_new_job       : chr  "1" ">4" "never" "never" ...
##  $ training_hours     : int  36 47 83 52 8 24 24 18 46 123 ...
##  $ target             : int  1 0 0 1 0 1 0 1 1 0 ...
```

Variable data types are shown above. We need to encode some variables before we perform any analysis.

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.0.3
```

```
#for relevent_experience
data$relevent_experience <- revalue(data$relevent_experience,
                                    c("Has relevent experience"=1))
data$relevent_experience <- revalue(data$relevent_experience,
                                    c("No relevent experience"=0))
data$relevent_experience <-as.numeric(data$relevent_experience)

#for gender
data$gender <-as.numeric(factor(data$gender, levels = c("Male","Female","Other")))

#for enrolled_university
data$enrolled_university <- revalue(data$enrolled_university,
                                    c("no_enrollment"=0))
data$enrolled_university <- revalue(data$enrolled_university,
                                    c("Part time course"=1))
data$enrolled_university <- revalue(data$enrolled_university,
                                    c("Full time course" = 2))
data$enrolled_university <-as.numeric(data$enrolled_university)

#for education_level
data$education_level <- as.numeric(factor(data$education_level,
                                          levels = c("Primary School",
                                    "High School","Graduate","Masters","Phd")))

#for experience
data$experience <- revalue(data$experience, c("<1"=0))
data$experience <- revalue(data$experience, c(">20"=21))
data$experience<-as.numeric(data$experience)

#for company size
data$company_size <- as.numeric(factor(data$company_size, levels = c("<10",
                                    "10/49","50-99" ,"100-500","500-999",
                                    "1000-4999","5000-9999", "10000+" )))

#for last_new_job
data$last_new_job <- revalue(data$last_new_job, c("never"=0))
data$last_new_job <- revalue(data$last_new_job, c(">4"=5))
data$last_new_job <-as.numeric(data$last_new_job)
```

Since city id and enrollee id are not needed in the analysis, I will drop these two variables. The complete variables are training_hours, relevent_experience and target. For categorical variables, gender has 4508 missing values, which is about 31% of the total observations.Since the only continuous variable with missing values (experience) has only 65 missing NA's, I will drop the 65 NA's for experience and generate 20% missing values for another continuous variable: training hours. So we have one complete and one 20% missing continuous variable.

```r
#keep the variables that can be used for analysis
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
data2 = select(data,-c('enrollee_id','city'))
#delete the 65 NA observations for experience
data2=data2[!is.na(data2$experience), ]
```

```r
#Generate missing values for training_hours depending on one variable
library(dplyr)
data_new = select(data2,'experience','training_hours')
library(mice)
```

```
## Warning: package 'mice' was built under R version 4.0.3

##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

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

```r
set.seed(102)
cont_cat = ampute(data_new,prop = 0.2,patterns=c(1,0),mech = "MAR")$amp
data2['training_hours'] = cont_cat['training_hours']
```

Listwise Imputation

Delete the entire row of the data if any variable has a missing value.

```r
#listwise deletion based on the original data set
data_complete = na.omit(data)
model1 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
+ +          enrolled_university+education_level+company_size+experience,
data = data_complete, family = binomial())
summary(model1)
```

```
##
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + ++enrolled_university + education_level +
##     company_size + experience, family = binomial(), data = data_complete)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9879  -0.6624  -0.5173  -0.3838   2.4586
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.4468840  0.1765428  -8.196 2.49e-16 ***
## training_hours      -0.0003550  0.0004637  -0.765 0.443974
## factor(gender)2     -0.1856524  0.0986021  -1.883 0.059721 .
## factor(gender)3     -0.3716035  0.2938973  -1.264 0.206087
## relevent_experience  0.0122720  0.0783949   0.157 0.875607
## last_new_job         0.0185103  0.0196855   0.940 0.347063
## enrolled_university  0.1356937  0.0403190   3.366 0.000764 ***
## education_level      0.1612483  0.0464179   3.474 0.000513 ***
## company_size         0.0188304  0.0126777   1.485 0.137461
## experience          -0.0846922  0.0053724 -15.764  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8931.2  on 10128  degrees of freedom
## Residual deviance: 8553.1  on 10119  degrees of freedom
## AIC: 8573.1
##
## Number of Fisher Scoring iterations: 5
```

Mean/Mode Imputation

For numeric/continuous variables, use the mean value for all missing values. For categorical variables, use the mode value for all missing values.

```r
#copy data set
data3 = data2
#Mean imputation for numeric missing values
#create function for mean imputation
mean.imp <- function (a)
{ missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
```

```r
  imputed[missing] <- mean(a.obs)
  return (imputed)
}
#impute
data3['training_hours']=mean.imp(data3['training_hours'])

#Mode imputation for the categorical variables
#create function for mode imputation
mode <- function (a)
{ ta =table(a)
  tam = max(ta)
  if(all(ta==tam))
    mod =NA
  else
    mod = as.numeric(names(ta)[ta==tam])
  return (mod)
}

mode.imp <- function (a)
{
  missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- mode(a.obs)
  return (imputed)
}

#impute
data3['enrolled_university'] = mode.imp(data3['enrolled_university'])
data3['education_level'] = mode.imp(data3['education_level'])
data3['company_size'] = mode.imp(data3['company_size'])
data3['last_new_job'] = mode.imp(data3['last_new_job'])
data3$gender = mode.imp(data3$gender)
```

```r
model2 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
              enrolled_university+education_level+company_size+experience,
             data = data3,family=binomial())
summary(model2)
```

```
##
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience, family = binomial(), data = data3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3006  -0.7865  -0.6367  -0.4193   2.2394
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.8941612  0.0920650  -9.712  < 2e-16 ***
## training_hours      -0.0009414  0.0003220  -2.923  0.00346 **
```

```
## factor(gender)2      0.0165932  0.0687499   0.241  0.80928
## factor(gender)3     -0.0322299  0.1717748  -0.188  0.85117
## relevent_experience -0.2839029  0.0408618  -6.948 3.71e-12 ***
## last_new_job         0.0231773  0.0124875   1.856  0.06345 .
## enrolled_university  0.2235477  0.0220260  10.149  < 2e-16 ***
## education_level      0.2197541  0.0266603   8.243  < 2e-16 ***
## company_size        -0.0753646  0.0096598  -7.802 6.10e-15 ***
## experience          -0.0552043  0.0032896 -16.782  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 20484  on 19083  degrees of freedom
## AIC: 20504
##
## Number of Fisher Scoring iterations: 4
```

```
mean(abs(model1$coef -model2$coef)/abs(model1$coef))
```

```
## [1] 3.478384
```

Random Imputation

Sample a random value from the non-missing observations of the variable for each missing value.

```
data4 = data2
#create random imputation function
random.imp <- function (a)
{
  missing <- is.na(a)
  n.missing <- sum(missing)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- as.numeric(sample (a.obs, n.missing, replace=TRUE))
  return (imputed)
}

#impute for each variable with missing values
data4$gender = random.imp(data2$gender)
data4$education_level = random.imp(data2$education_level)
data4$company_size = random.imp(data2$company_size)
data4$last_new_job = random.imp(data2$last_new_job)
data4$enrolled_university= random.imp(data2$enrolled_university)
data4$training_hours = random.imp(data2$training_hours)
```

```
model3 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
              enrolled_university+education_level+company_size+experience,
             family=binomial(),data = data4)
summary(model3)
```

```
##
```

```
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience, family = binomial(), data = data4)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2768  -0.7843  -0.6419  -0.4446   2.2186
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.0951361  0.0923388 -11.860  < 2e-16 ***
## training_hours      -0.0008538  0.0002925  -2.918  0.00352 **
## factor(gender)2      0.0706602  0.0599485   1.179  0.23853
## factor(gender)3     -0.0903486  0.1519312  -0.595  0.55207
## relevent_experience -0.2911372  0.0408024  -7.135 9.66e-13 ***
## last_new_job         0.0213501  0.0121900   1.751  0.07987 .
## enrolled_university  0.2285010  0.0219094  10.429  < 2e-16 ***
## education_level      0.1941359  0.0261341   7.428 1.10e-13 ***
## company_size        -0.0001178  0.0078631  -0.015  0.98805
## experience          -0.0562897  0.0032736 -17.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 20548  on 19083  degrees of freedom
## AIC: 20568
##
## Number of Fisher Scoring iterations: 4
```

```
mean(abs(model1$coef -model3$coef)/abs(model1$coef))
```

```
## [1] 3.089242
```

Dummy Variable Imputation Use some arbitrary number for the missing values (i.e. mean for numeric, mode for categorical) and add a dummy variable as an indicator for missing-ness.

```
#copy data set
data5 = data2
#dummy variable imputation on training_hours
data5['training_hours']=mean.imp(data5['training_hours'])
d1 = is.na(data2$training_hours)

#dummy variable imputation on other categorical variables
data5$gender = mode.imp(data5$gender)
data5['education_level'] = mode.imp(data5['education_level'])
data5['company_size'] = mode.imp(data5['company_size'])
data5['last_new_job'] = mode.imp(data5['last_new_job'])
data5['enrolled_university'] = mode.imp(data5['enrolled_university'])
d2 = is.na(data2$gender)
d3 = is.na(data2$education_level)
```

```r
d4 = is.na(data2$company_size)
d5 = is.na(data2$last_new_job)
d6 = is.na(data2$enrolled_university)
```

```r
model4 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
    enrolled_university+education_level+company_size+experience+d1+d2+d3+d4+d5+d6,
             data = data5,family=binomial())
summary(model4)
```

```
##
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience + d1 + d2 + d3 + d4 + d5 + d6, family = binomial(),
##     data = data5)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5972  -0.7422  -0.5831  -0.3216   2.4605
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.2035733  0.1065061 -20.690  < 2e-16 ***
## training_hours       -0.0008726  0.0003319  -2.629 0.008559 **
## factor(gender)2       0.0906133  0.0716633   1.264 0.206076
## factor(gender)3      -0.0024921  0.1769025  -0.014 0.988760
## relevent_experience   0.1715187  0.0460579   3.724 0.000196 ***
## last_new_job          0.0600381  0.0128297   4.680 2.87e-06 ***
## enrolled_university   0.1586432  0.0231293   6.859 6.94e-12 ***
## education_level       0.3078514  0.0274193  11.228  < 2e-16 ***
## company_size          0.0027992  0.0106227   0.264 0.792155
## experience           -0.0682503  0.0036281 -18.812  < 2e-16 ***
## d1TRUE                0.0792810  0.0495959   1.599 0.109923
## d2TRUE                0.2135221  0.0419846   5.086 3.66e-07 ***
## d3TRUE               -0.6904269  0.1233371  -5.598 2.17e-08 ***
## d4TRUE                1.2032592  0.0431853  27.863  < 2e-16 ***
## d5TRUE                0.1181799  0.1143934   1.033 0.301556
## d6TRUE                0.1095881  0.1212057   0.904 0.365916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 19637  on 19077  degrees of freedom
## AIC: 19669
##
## Number of Fisher Scoring iterations: 4
```

```r
mean(abs(model1$coef -model4$coef[1:10])/abs(model1$coef))
```

```
## [1] 2.180634
```

Hotdecking Imputation

For each variables with missing values, the complete set of the data is compared with the missing set. The missing values are filled with the nearest distanced non-missing values.

```
library(VIM)
```

```
## Warning: package 'VIM' was built under R version 4.0.4

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
data_h = data2
data_h = kNN(data_h,k=1,imp_var=F)
```

```
model5 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
              enrolled_university+education_level+company_size+experience,
             data = data_h,family=binomial())
summary(model5)
```

```
##
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience, family = binomial(), data = data_h)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2829  -0.7815  -0.6399  -0.4366   2.2255
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.2593592  0.0909895 -13.841  < 2e-16 ***
## training_hours      -0.0007709  0.0002875  -2.682  0.00732 **
## factor(gender)2      0.0455588  0.0600683   0.758  0.44818
## factor(gender)3     -0.0559360  0.1487717  -0.376  0.70693
## relevent_experience -0.2863782  0.0411414  -6.961 3.38e-12 ***
## last_new_job         0.0136526  0.0123591   1.105  0.26931
## enrolled_university  0.2420505  0.0219444  11.030  < 2e-16 ***
```

```
## education_level      0.2231562  0.0261986   8.518  < 2e-16 ***
## company_size         0.0186099  0.0078433   2.373  0.01766 *
## experience          -0.0567940  0.0033123 -17.146  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 20509  on 19083  degrees of freedom
## AIC: 20529
##
## Number of Fisher Scoring iterations: 4
```

```r
mean(abs(model1$coef -model5$coef)/abs(model1$coef))
```

```
## [1] 2.950348
```

Regression Imputation

First fit a regression model with the variable with missing values as the outcome variable and the complete variables as covariates. Then use the model to compute for missing values.

```r
#copy data set
data6 = data2
#variables without missing values are: target,experience and relevent_experience
#Missing data indicator
Ry = as.numeric(!is.na(data6$training_hours))
data.cc = data6[Ry ==1, ]
data.dropped = data6[Ry ==0, ]
reg = lm(training_hours ~relevent_experience+target+experience,data = data.cc)
y.imp = predict(reg, newdata = data.dropped)
data6$training_hours[Ry == 0] = y.imp
```

```r
#for categorical variables
#select the complete variables
x<- select(data6,'relevent_experience','target','experience')

#use polytomous regression for categorical variables that are not dichotomous
Ry1 = as.numeric(!is.na(data6$gender))
gender.imp = mice.impute.polyreg(data6$gender, !is.na(data6$gender), x)
# Impute the predictions where they belong:
data6$gender[Ry1 == 0] = gender.imp

#use another function from the mice package for ordered categorical variables
Ry2 = as.numeric(!is.na(data6$education_level))
edu.imp = mice.impute.polr(data6$education_level, !is.na(data6$education_level), x)
# Impute the predictions where they belong:
data6$education_level[Ry2 == 0] = as.numeric(edu.imp)

Ry3 = as.numeric(!is.na(data6$enrolled_university))
enu.imp = mice.impute.polr(data6$enrolled_university, !is.na(data6$enrolled_university), x)
# Impute the predictions where they belong:
```

```r
data6$enrolled_university[Ry3 == 0] = as.numeric(enu.imp)

Ry4 = as.numeric(!is.na(data6$company_size))
comp.imp = mice.impute.polr(data6$company_size, !is.na(data6$company_size), x)
# Impute the predictions where they belong:
data6$company_size[Ry4 == 0] =as.numeric(comp.imp)

Ry5 = as.numeric(!is.na(data6$last_new_job))
lnj.imp = mice.impute.polr(data6$last_new_job, !is.na(data6$last_new_job), x)
# Impute the predictions where they belong:
data6$last_new_job[Ry5 == 0] = as.numeric(lnj.imp)
```

```r
#the missing values in the categorical variables are omitted for the analysis
model6 = glm(target ~ training_hours+factor(gender)+relevent_experience+last_new_job+
              enrolled_university+education_level+company_size+experience,
             data = data6,family=binomial())
summary(model6)
```

```
##
## Call:
## glm(formula = target ~ training_hours + factor(gender) + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience, family = binomial(), data = data6)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2923  -0.7853  -0.6403  -0.4333   2.2170
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.1977270  0.0923984 -12.963  < 2e-16 ***
## training_hours      -0.0010962  0.0003231  -3.393 0.000692 ***
## factor(gender)2      0.0748944  0.0585978   1.278 0.201211
## factor(gender)3     -0.0088117  0.1452537  -0.061 0.951627
## relevent_experience -0.2803870  0.0411359  -6.816 9.35e-12 ***
## last_new_job         0.0157746  0.0123900   1.273 0.202956
## enrolled_university  0.2264022  0.0220060  10.288  < 2e-16 ***
## education_level      0.2021506  0.0261860   7.720 1.17e-14 ***
## company_size         0.0234181  0.0077400   3.026 0.002481 **
## experience          -0.0569248  0.0033145 -17.174  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 20532  on 19083  degrees of freedom
## AIC: 20552
##
## Number of Fisher Scoring iterations: 4
```

```r
mean(abs(model1$coef -model6$coef)/abs(model1$coef))
```

```
## [1] 3.012915
```

Regression with Noise

Base on regression imputation, add a noise term to the regression model.

```r
#we don't have dichotomous categorical variables
#so we only add noise to the numeric variable
data7 = data6
data7$training_hours = data2$training_hours
noise = rnorm(length(y.imp), 0, summary(reg)$sigma)
y.imps = y.imp + noise
data7$training_hours[Ry == 0] = y.imps
```

```r
#the missing values in the categorical variables are omitted for the analysis
model7 = glm(target ~ training_hours+gender+relevent_experience+last_new_job+
              enrolled_university+education_level+company_size+experience,
             data = data7,family=binomial())
summary(model7)
```

```
##
## Call:
## glm(formula = target ~ training_hours + gender + relevent_experience +
##     last_new_job + enrolled_university + education_level + company_size +
##     experience, family = binomial(), data = data7)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2921  -0.7848  -0.6404  -0.4283   2.2033
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.1984263  0.0918976 -13.041  < 2e-16 ***
## training_hours      -0.0010954  0.0002922  -3.749 0.000178 ***
## gender2              0.0745993  0.0586030   1.273 0.203032
## gender3             -0.0086385  0.1452701  -0.059 0.952581
## relevent_experience -0.2803760  0.0411385  -6.815 9.40e-12 ***
## last_new_job         0.0159173  0.0123901   1.285 0.198904
## enrolled_university  0.2262895  0.0220088  10.282  < 2e-16 ***
## education_level      0.2025468  0.0261855   7.735 1.03e-14 ***
## company_size         0.0233081  0.0077410   3.011 0.002604 **
## experience          -0.0569953  0.0033150 -17.193  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21431  on 19092  degrees of freedom
## Residual deviance: 20529  on 19083  degrees of freedom
## AIC: 20549
##
## Number of Fisher Scoring iterations: 4
```

```r
mean(abs(model1$coef -model7$coef)/abs(model1$coef))
```

```
## [1] 3.011157
```

Multiple Imputation using MI package

Creating multiple imputed data sets. Imputation methods are indicated in the table provided by the mi package.

```r
library(mi)
```

```
## Warning: package 'mi' was built under R version 4.0.3

## Loading required package: Matrix

## Loading required package: stats4

## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

## mi (Version 1.0, packaged: 2015-04-16 14:03:10 UTC; goodrich)

## mi  Copyright (C) 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015 Trustees of Columbia University

## This program comes with ABSOLUTELY NO WARRANTY.

## This is free software, and you are welcome to redistribute it

## under the General Public License version 2 or later.

## Execute RShowDoc('COPYING') for details.

##
## Attaching package: 'mi'

## The following objects are masked from 'package:mice':
##
##     complete, pool
```

```r
# Create the missing data frame object
mdf = missing_data.frame(data2)

# Examine the default settings
show(mdf)
```

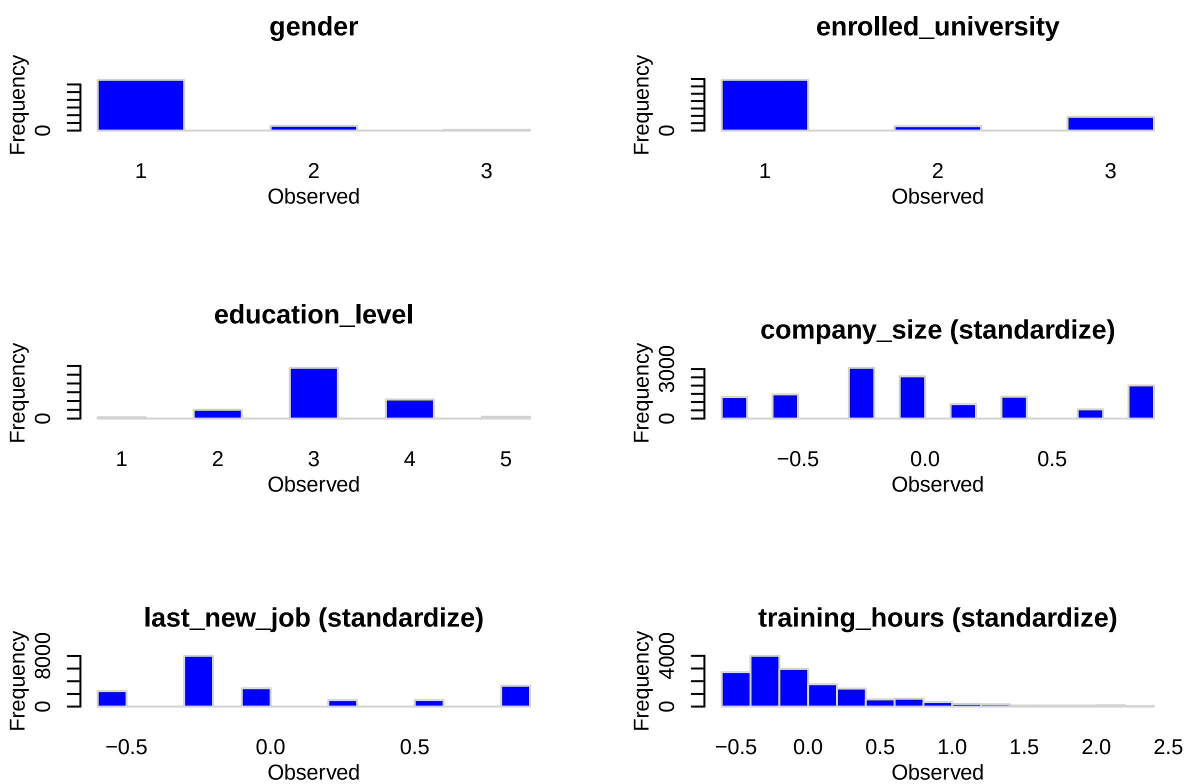
```
## Object of class missing_data.frame with 19093 observations on 9 variables
##
## There are 56 missing data patterns
##
## Append '@patterns' to this missing_data.frame to access the corresponding pattern for every observati
##
##                                    type missing method  model
## gender              ordered-categorical    4459    ppd ologit
## relevent_experience              binary       0   <NA>   <NA>
## enrolled_university ordered-categorical     381    ppd ologit
## education_level     ordered-categorical     450    ppd ologit
## experience                   continuous       0   <NA>   <NA>
## company_size                 continuous    5915    ppd linear
## last_new_job                 continuous     399    ppd linear
## training_hours               continuous    3905    ppd linear
## target                           binary       0   <NA>   <NA>
##
##                          family     link transformation
## gender              multinomial    logit           <NA>
## relevent_experience        <NA>     <NA>           <NA>
## enrolled_university multinomial    logit           <NA>
## education_level     multinomial    logit           <NA>
## experience                 <NA>     <NA>    standardize
## company_size           gaussian identity    standardize
## last_new_job           gaussian identity    standardize
## training_hours         gaussian identity    standardize
## target                     <NA>     <NA>           <NA>
```

```r
# Five-number summary statistics + missing number
summary(mdf)
```

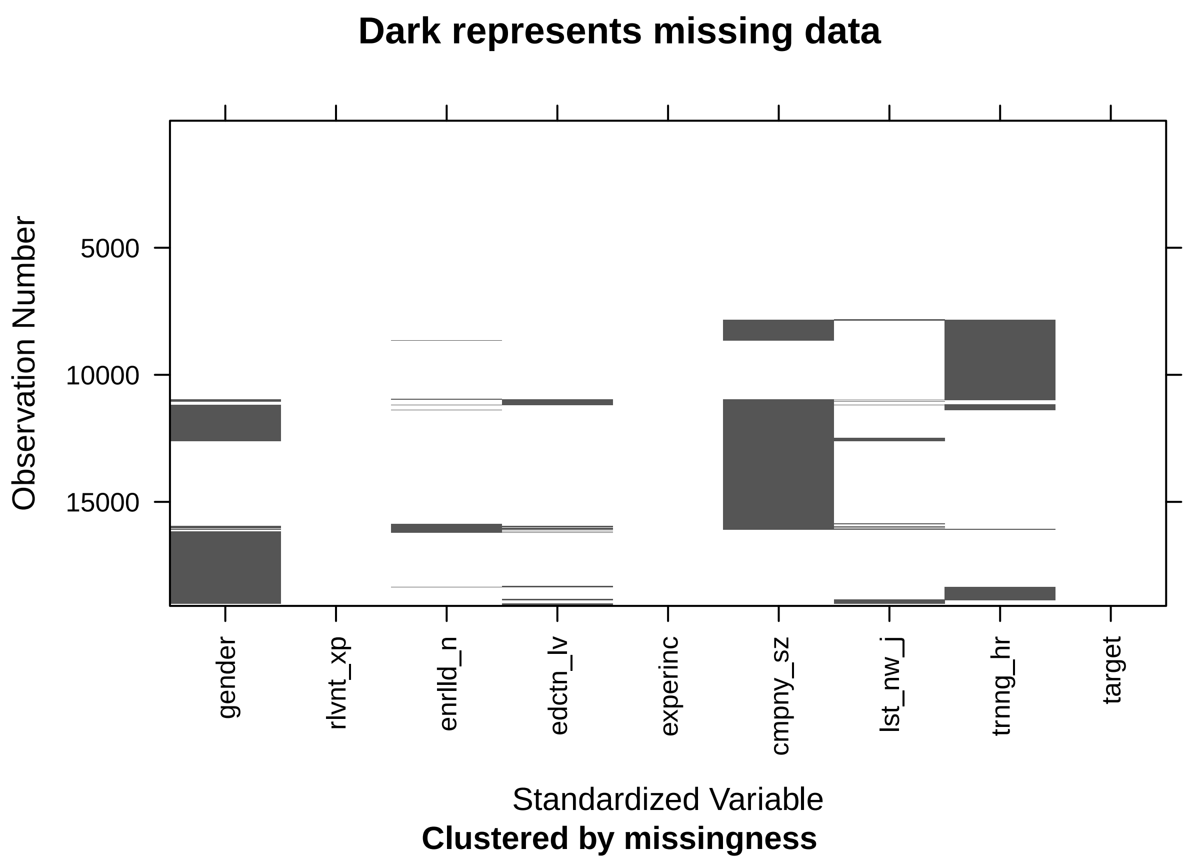
```
##      gender     relevent_experience enrolled_university education_level
##  Min.   :1.00   Min.   :0.0000      Min.   :0.0000      Min.   :1.000
##  1st Qu.:1.00   1st Qu.:0.0000      1st Qu.:0.0000      1st Qu.:3.000
##  Median :1.00   Median :1.0000      Median :0.0000      Median :3.000
##  Mean   :1.11   Mean   :0.7201      Mean   :0.4634      Mean   :3.136
##  3rd Qu.:1.00   3rd Qu.:1.0000      3rd Qu.:1.0000      3rd Qu.:4.000
##  Max.   :3.00   Max.   :1.0000      Max.   :2.0000      Max.   :5.000
##  NA's   :4459                       NA's   :381         NA's   :450
##    experience    company_size    last_new_job   training_hours
##  Min.   : 0.0   Min.   :1.000   Min.   :0.000   Min.   :  1.00
##  1st Qu.: 4.0   1st Qu.:3.000   1st Qu.:1.000   1st Qu.: 23.00
##  Median : 9.0   Median :4.000   Median :1.000   Median : 47.00
##  Mean   :10.1   Mean   :4.252   Mean   :2.001   Mean   : 65.34
##  3rd Qu.:16.0   3rd Qu.:6.000   3rd Qu.:3.000   3rd Qu.: 88.00
##  Max.   :21.0   Max.   :8.000   Max.   :5.000   Max.   :336.00
##                 NA's   :5915    NA's   :399     NA's   :3905
##      target
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.249
##  3rd Qu.:0.000
##  Max.   :1.000
```

```
##
```

```r
# Histograms of all variables with missing values
hist(mdf)
```




```r
# Graph of the missing pattern matrix R
image(mdf, grayscale=TRUE)
```




```r
mdf <- change(mdf, y = "last_new_job", what = "type", to = "ordered-categorical")
mdf <- change(mdf, y = "company_size", what = "type", to = "ordered-categorical")
mdf <- change(mdf, y = "gender", what = "type", to = "unorder")
mdf <- change(mdf, y = "training_hours", what = "type", to = "pos")
show(mdf)
```

```
## Object of class missing_data.frame with 19093 observations on 9 variables
##
## There are 56 missing data patterns
##
## Append '@patterns' to this missing_data.frame to access the corresponding pattern for every observati
##
##                                      type missing method  model
## gender                unordered-categorical    4459    ppd mlogit
## relevent_experience                  binary       0   <NA>   <NA>
## enrolled_university   ordered-categorical     381    ppd ologit
## education_level       ordered-categorical     450    ppd ologit
## experience                       continuous       0   <NA>   <NA>
## company_size          ordered-categorical    5915    ppd ologit
## last_new_job          ordered-categorical     399    ppd ologit
## training_hours        positive-continuous    3905    ppd linear
## target                               binary       0   <NA>   <NA>
##
##                          family     link transformation
## gender                multinomial    logit           <NA>
## relevent_experience          <NA>     <NA>           <NA>
```

```
## enrolled_university multinomial    logit          <NA>
## education_level     multinomial    logit          <NA>
## experience                 <NA>     <NA>   standardize
## company_size        multinomial    logit          <NA>
## last_new_job        multinomial    logit          <NA>
## training_hours         gaussian identity           log
## target                     <NA>     <NA>          <NA>
```

```
#Run mi with 5 chains and 50 iterations on the dataset
# Running the chains
imputations <- mi(mdf, n.chains = 5, n.iter=50)
```

```
#Check convergence/diagnostics and make changes if necessary
round(mipply(imputations, mean, to.matrix = TRUE), 3)
```

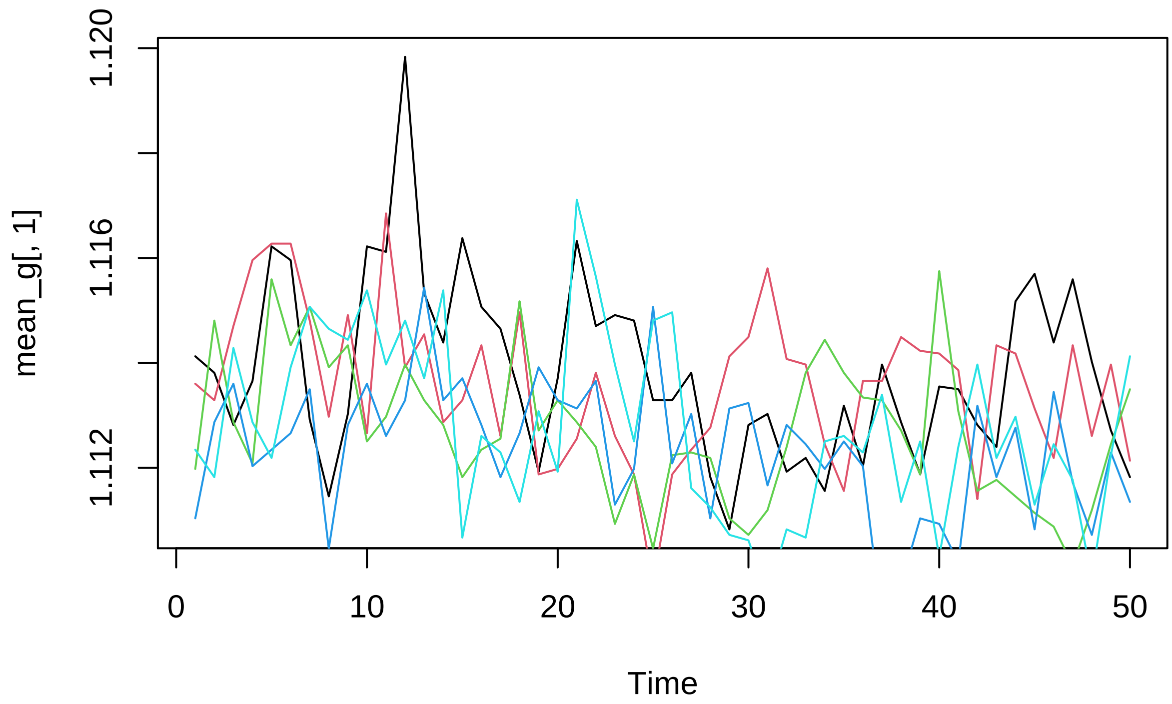
```
##                             chain:1 chain:2 chain:3 chain:4 chain:5
## gender                        1.112   1.112   1.113   1.111   1.114
## relevent_experience           1.720   1.720   1.720   1.720   1.720
## enrolled_university           1.468   1.468   1.468   1.468   1.469
## education_level               3.130   3.132   3.129   3.130   3.131
## experience                    0.000   0.000   0.000   0.000   0.000
## company_size                  4.229   4.224   4.227   4.216   4.200
## last_new_job                  2.989   2.989   2.990   2.988   2.988
## training_hours                3.762   3.759   3.763   3.759   3.761
## target                        1.249   1.249   1.249   1.249   1.249
## missing_gender                0.234   0.234   0.234   0.234   0.234
## missing_enrolled_university   0.020   0.020   0.020   0.020   0.020
## missing_education_level       0.024   0.024   0.024   0.024   0.024
## missing_company_size          0.310   0.310   0.310   0.310   0.310
## missing_last_new_job          0.021   0.021   0.021   0.021   0.021
## missing_training_hours        0.205   0.205   0.205   0.205   0.205
```

```
converged <- mi2BUGS(imputations)
Rhats(imputations)
```

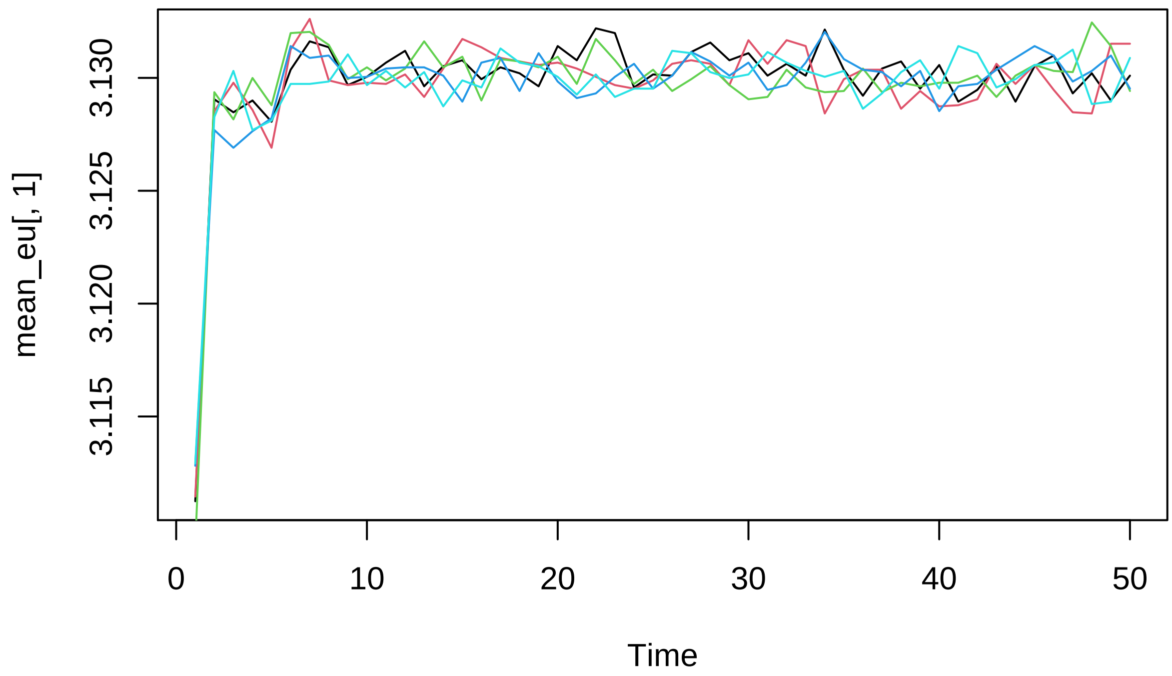
```
##              mean_gender mean_enrolled_university     mean_education_level
##                1.0756391                0.9934142                0.9904866
##        mean_company_size        mean_last_new_job      mean_training_hours
##                0.9924057                0.9926196                0.9970649
##                sd_gender   sd_enrolled_university       sd_education_level
##                1.0710602                0.9915491                0.9900725
##          sd_company_size          sd_last_new_job        sd_training_hours
##                0.9904997                0.9962266                1.0064827
```

The mean of each variable for each chain are roughly the same. The r hats are close to one.

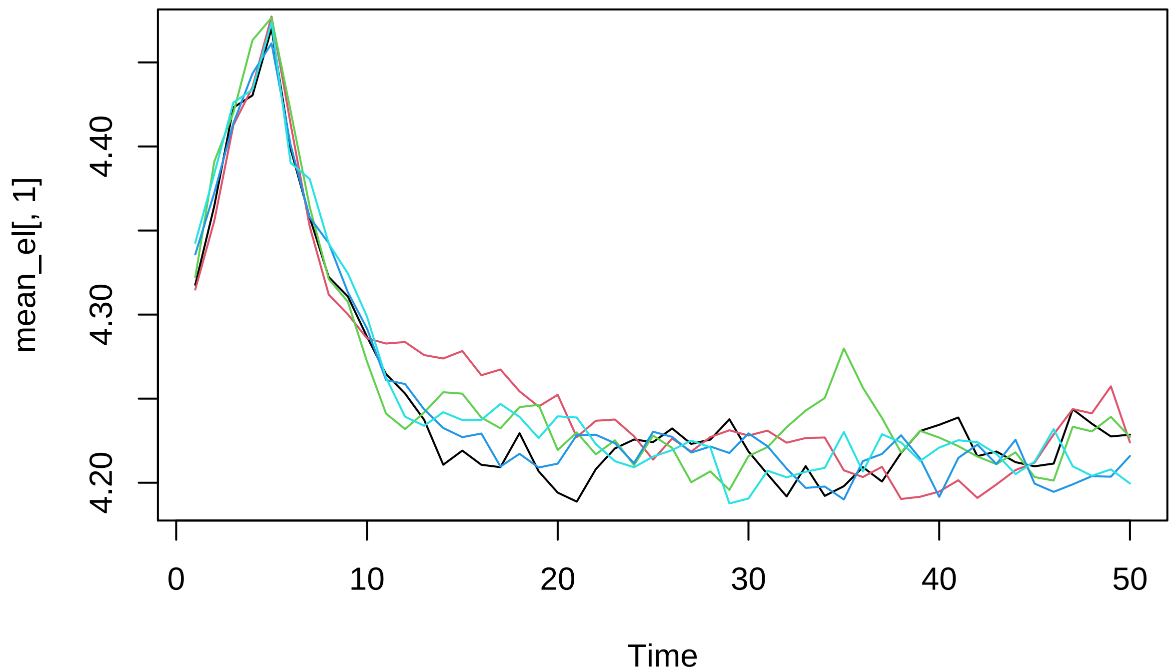
```
mean_g = converged[, , 1]
# Traceplot of mean imputed training hours
ts.plot(mean_g[,1], col=1)
lines(mean_g[,2], col= 2)
lines(mean_g[,3], col= 3)
lines(mean_g [,4], col= 4)
lines(mean_g [,5], col= 5)
```



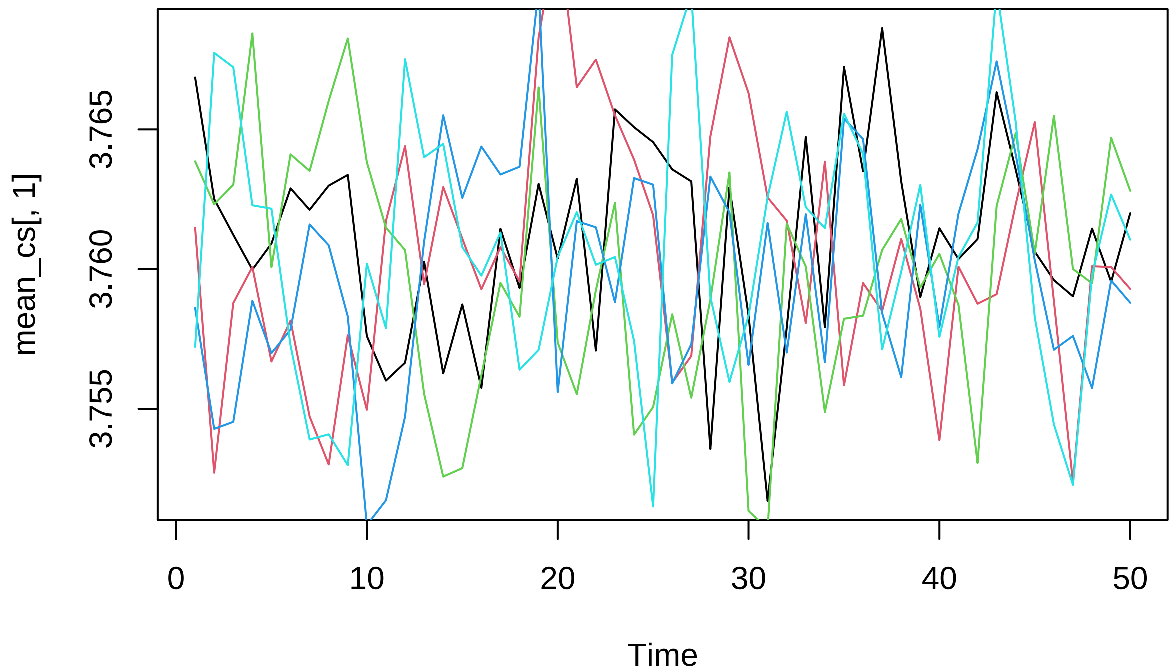

```
mean_eu = converged[, , 3]
# Traceplot of mean imputed last new job
ts.plot(mean_eu[,1], col=1)
lines(mean_eu[,2], col= 2)
lines(mean_eu[,3], col= 3)
lines(mean_eu [,4], col= 4)
lines(mean_eu[,5], col= 5)
```



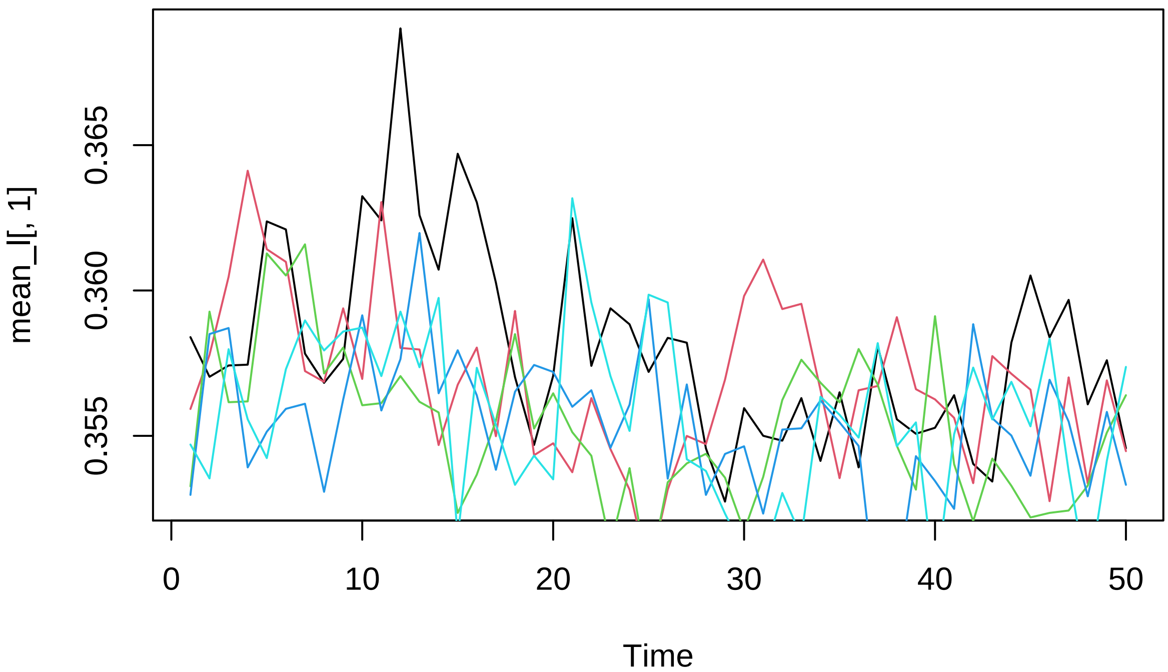

```
mean_el = converged[, , 4]
# Traceplot of mean imputed last new job
ts.plot(mean_el[,1], col=1)
lines(mean_el[,2], col= 2)
lines(mean_el[,3], col= 3)
lines(mean_el [,4], col= 4)
lines(mean_el [,5], col= 5)
```



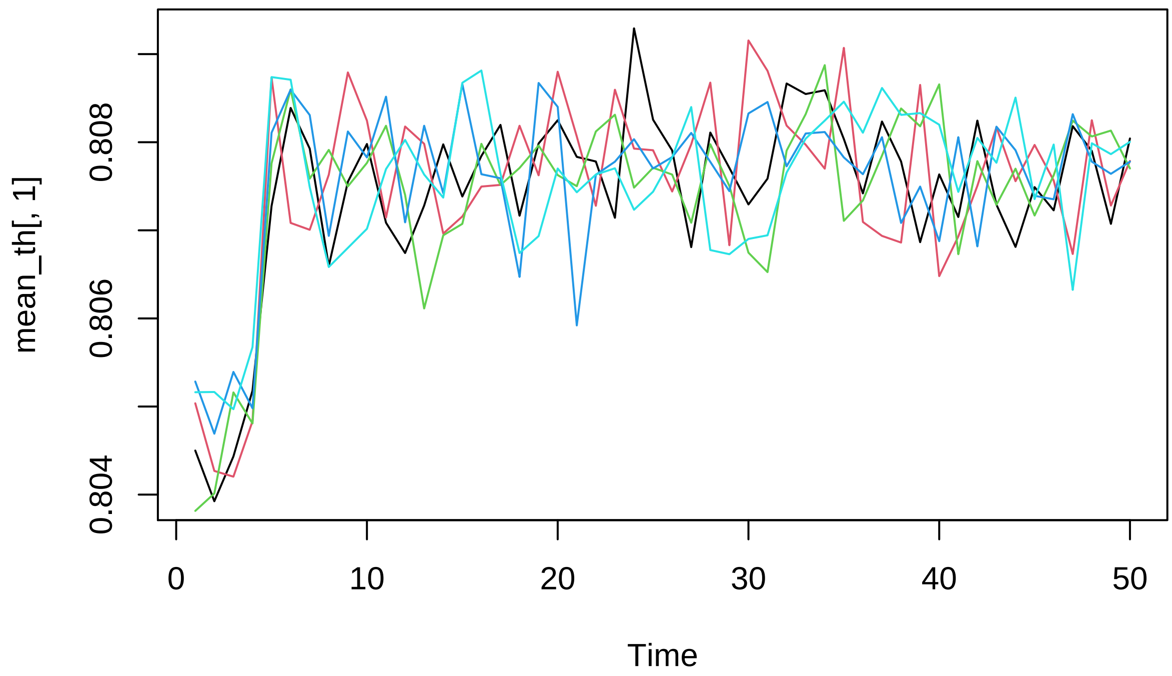

```
mean_cs = converged[, , 6]
# Traceplot of mean imputed last new job
ts.plot(mean_cs[,1], col=1)
lines(mean_cs[,2], col= 2)
lines(mean_cs[,3], col= 3)
lines(mean_cs [,4], col= 4)
lines(mean_cs[,5], col= 5)
```



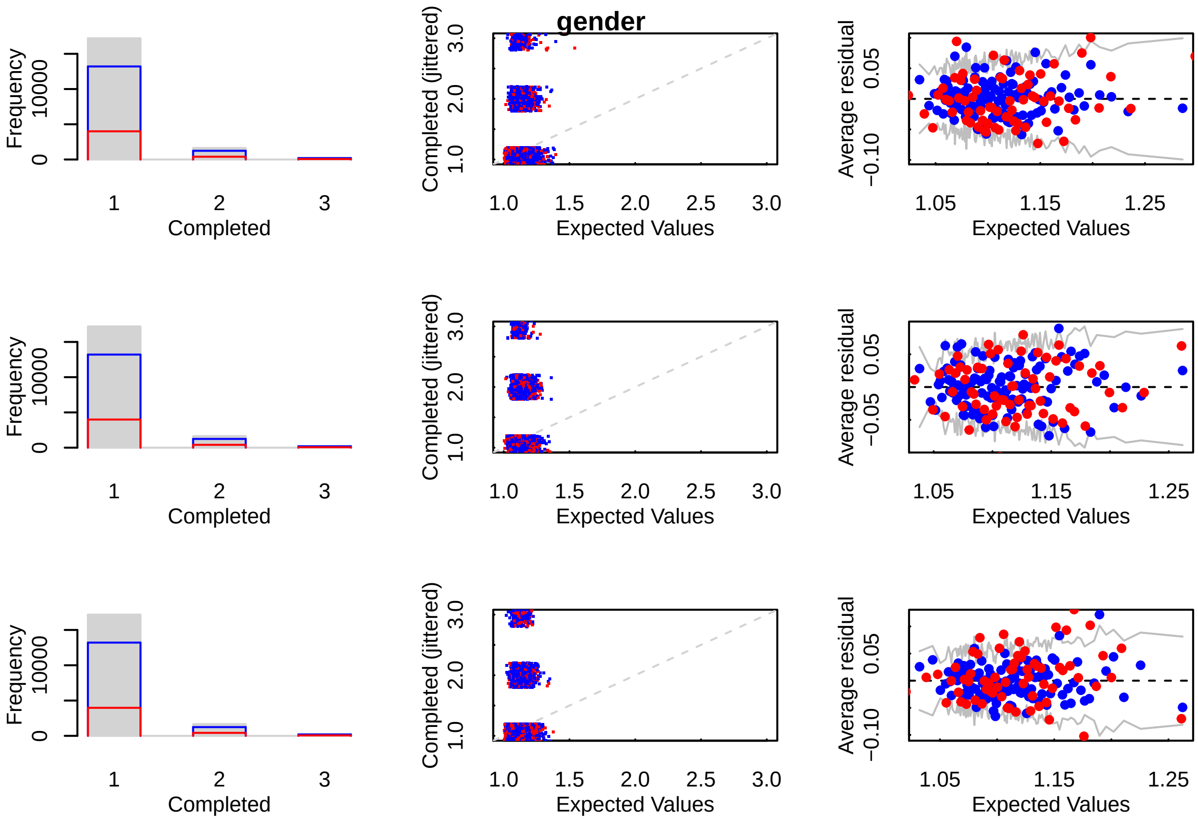

```
mean_l = converged[, , 7]
# Traceplot of mean imputed last new job
ts.plot(mean_l[,1], col=1)
lines(mean_l[,2], col= 2)
lines(mean_l[,3], col= 3)
lines(mean_l [,4], col= 4)
lines(mean_l [,5], col= 5)
```

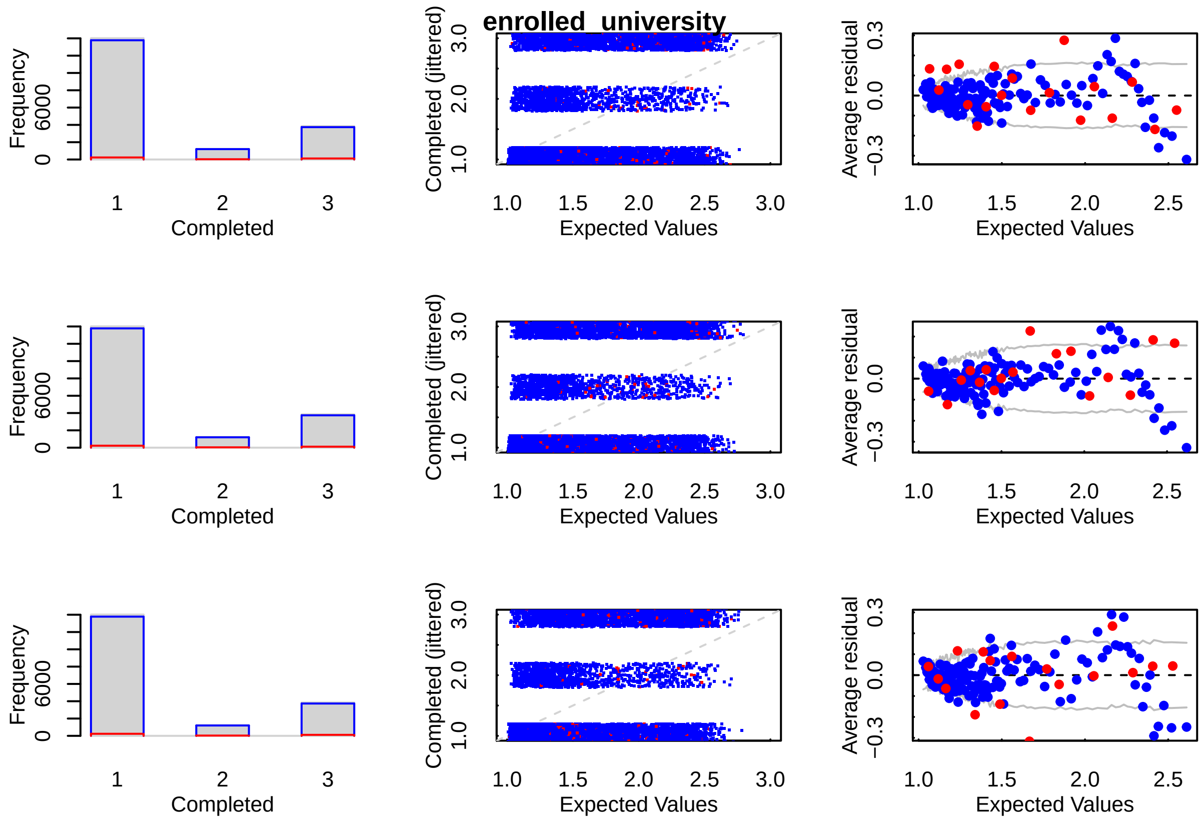


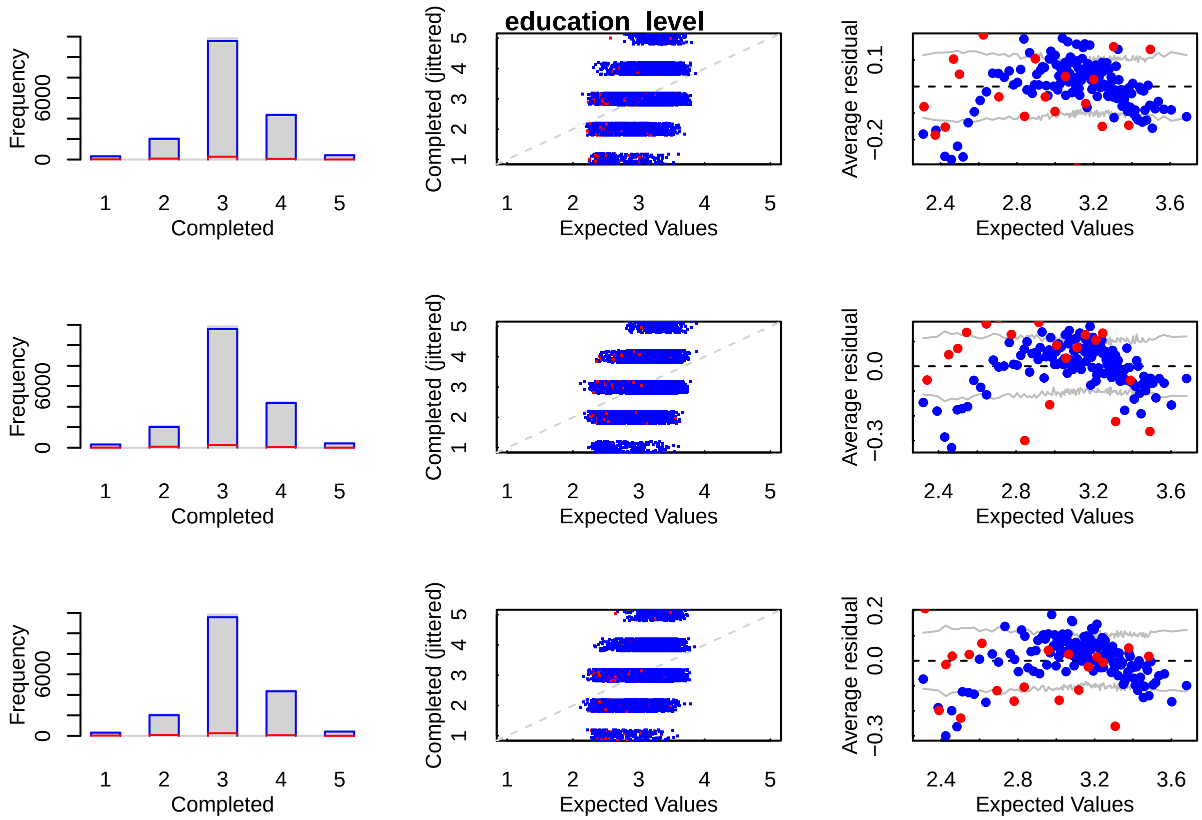

```r
mean_th = converged[, , 8]
# Traceplot of mean imputed last new job
ts.plot(mean_th[,1], col=1)
lines(mean_th[,2], col= 2)
lines(mean_th[,3], col= 3)
lines(mean_th [,4], col= 4)
lines(mean_th[,5], col= 5)
```

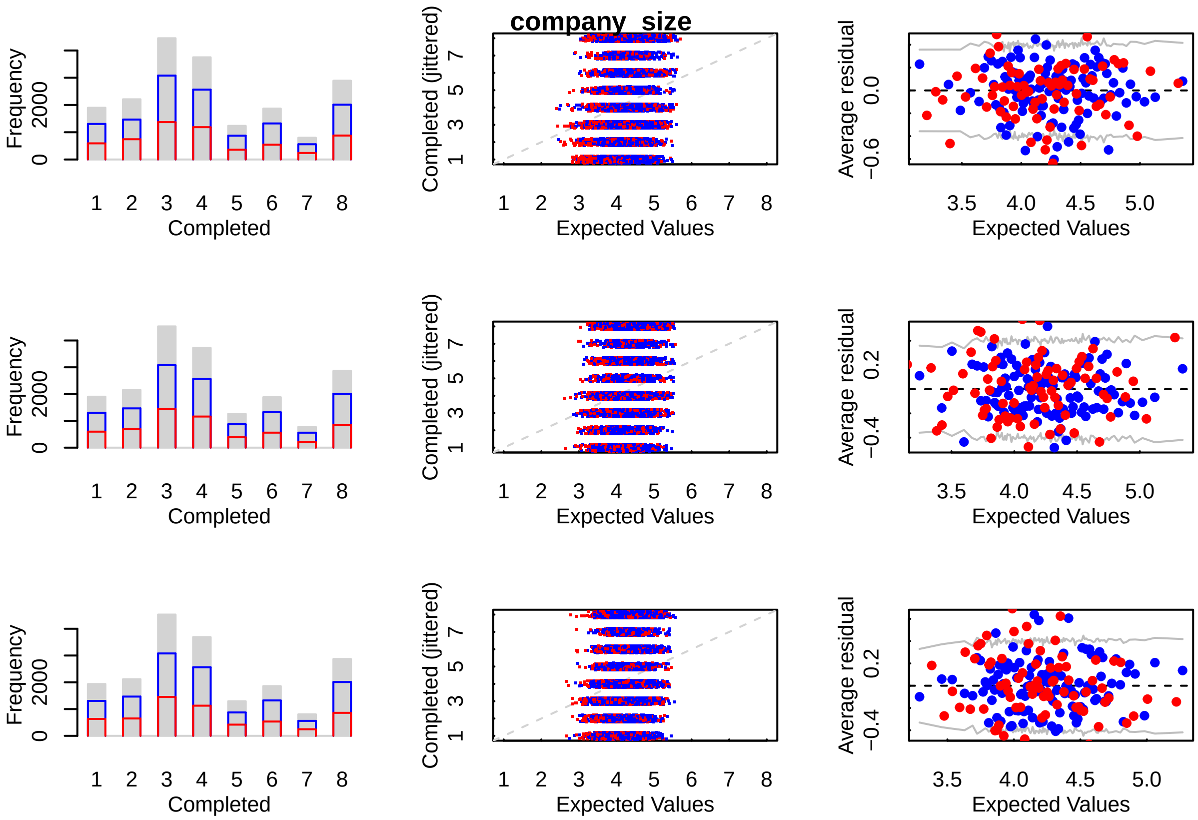



```r
plot(imputations)
```

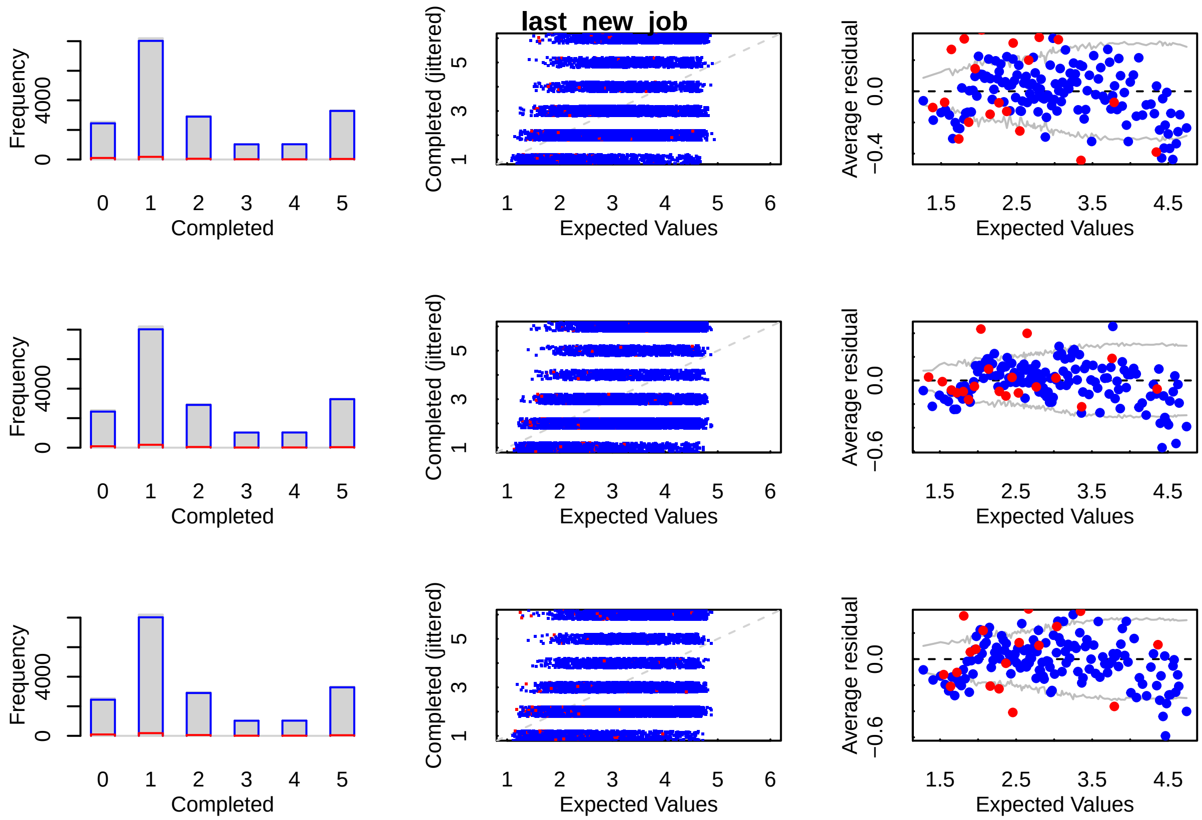

gender
Frequency
0
10000
1
2
3
Completed
Completed (jittered)
1.0
2.0
3.0
1.0
1.5
2.0
2.5
3.0
Expected Values
Average residual
−0.10
0.05
1.05
1.15
1.25
Expected Values
Frequency
0
10000
1
2
3
Completed
Completed (jittered)
1.0
2.0
3.0
1.0
1.5
2.0
2.5
3.0
Expected Values
Average residual
−0.05
0.05
1.05
1.15
1.25
Expected Values
Frequency
0
10000
1
2
3
Completed
Completed (jittered)
1.0
2.0
3.0
1.0
1.5
2.0
2.5
3.0
Expected Values
Average residual
−0.10
0.05
1.05
1.15
1.25
Expected Values

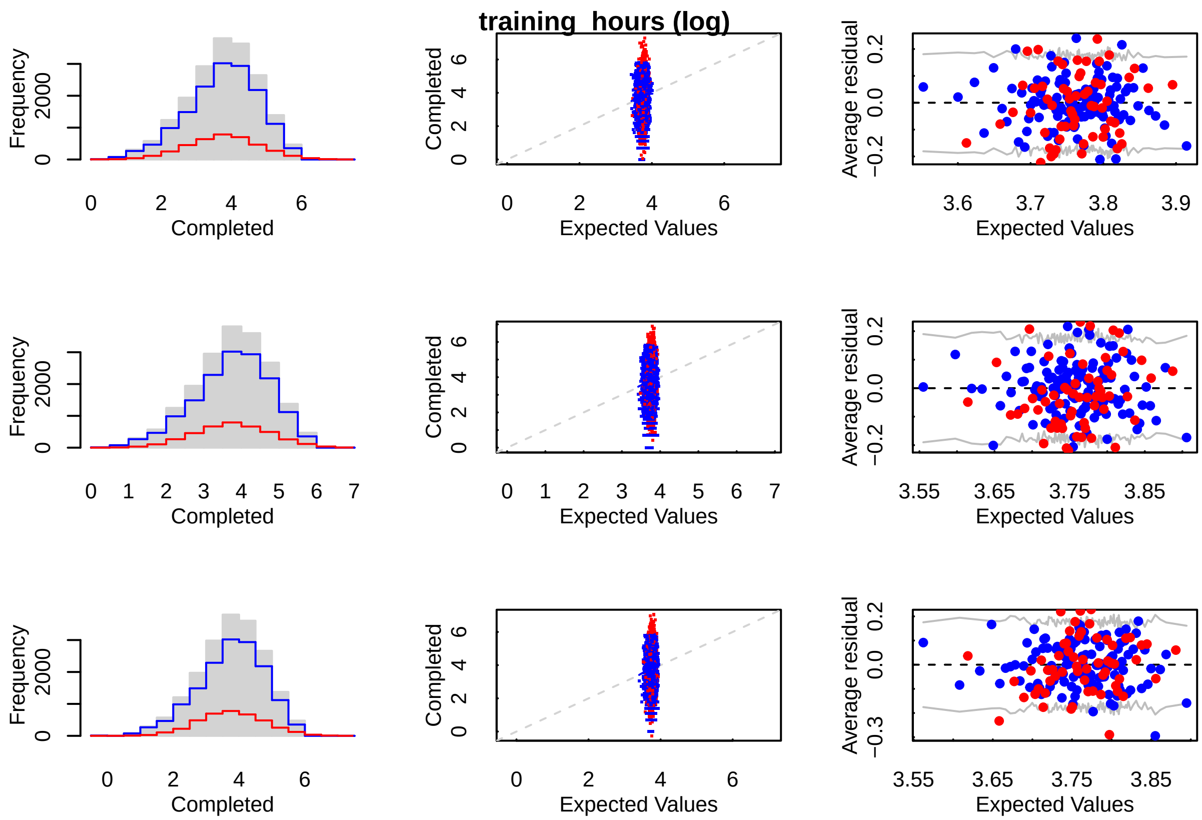

enrolled university
Frequency
0
6000
1
2
3
Completed
Completed (jittered)
1.0
2.0
3.0
1.0
1.5
2.0
2.5
3.0
Expected Values
Average residual
−0.3
0.0
0.3
1.0
1.5
2.0
2.5
Expected Values


education_level
Frequency
6000
0
1
2
3
4
5
Completed
Completed (jittered)
Expected Values
Average residual
0.1
−0.2
2.4
2.8
3.2
3.6
0.0
−0.3
0.2



company_size
Frequency
Completed
Completed (jittered)
Expected Values
Average residual


last new job
Frequency
0
4000
0 1 2 3 4 5
Completed
Completed (jittered)
1
3
5
1 2 3 4 5 6
Expected Values
Average residual
−0.4
0.0
1.5 2.5 3.5 4.5
Expected Values
Frequency
0
4000
0 1 2 3 4 5
Completed
Completed (jittered)
1
3
5
1 2 3 4 5 6
Expected Values
Average residual
−0.6
0.0
1.5 2.5 3.5 4.5
Expected Values
Frequency
0
4000
0 1 2 3 4 5
Completed
Completed (jittered)
1
3
5
1 2 3 4 5 6
Expected Values
Average residual
−0.6
0.0
1.5 2.5 3.5 4.5
Expected Values



The imputation converges, so the number of iteration is sufficient.

```
#Pool the results and report the estimated equation
model8 = mi::pool(target ~ training_hours+factor(gender)+relevent_experience+as.numeric(last_new_job)+
as.numeric(enrolled_university)+as.numeric(education_level)+
  as.numeric(company_size)+experience,family=binomial(),imputations)
display(model8)
```

```
## bayesglm(formula = target ~ training_hours + factor(gender) +
##     relevent_experience + as.numeric(last_new_job) + as.numeric(enrolled_university) +
##     as.numeric(education_level) + as.numeric(company_size) +
##     experience, data = imputations, family = binomial())
##                                 coef.est coef.se
## (Intercept)                     -1.46     0.11
## training_hours                   0.00     0.00
## factor(gender)2                  0.07     0.06
## factor(gender)3                  0.02     0.20
## relevent_experience1            -0.29     0.04
## as.numeric(last_new_job)         0.02     0.01
## as.numeric(enrolled_university)  0.23     0.02
## as.numeric(education_level)      0.21     0.03
## as.numeric(company_size)         0.01     0.01
## experience                      -0.06     0.00
## n = 19083, k = 10
## residual deviance = 20526.9, null deviance = 21430.9 (difference = 904.0)
```

```r
mean(abs(model1$coef - coef(model8))/abs(model1$coef))
```

```
## [1] 2.950267
```