

4.4_Mice_imputation_comb.rmd

Fay

2022-11-01

Load libraries

```
library(mice)
```

```
##  
## Attaching package: 'mice'  
  
## The following object is masked from 'package:stats':  
##  
##   filter  
  
## The following objects are masked from 'package:base':  
##  
##   cbind, rbind
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.1
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.1
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
```

```
## v ggplot2 3.3.6      v dplyr   1.0.10  
## v tibble  3.1.8      v stringr 1.4.1  
## v readr   2.1.3      v forcats 0.5.2  
## v purrr   0.3.5
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'readr' was built under R version 4.2.1
```

```
## Warning: package 'purrr' was built under R version 4.2.1
```

```

## Warning: package 'dplyr' was built under R version 4.2.1

## Warning: package 'stringr' was built under R version 4.2.1

## Warning: package 'forcats' was built under R version 4.2.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks mice::filter(), stats::filter()
## x dplyr::lag()     masks stats::lag()

library(VIM)

## Warning: package 'VIM' was built under R version 4.2.1

## 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

library(fitdistrplus)

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.2.1

##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: survival

## Warning: package 'survival' was built under R version 4.2.1

library(fitur)

## Warning: package 'fitur' was built under R version 4.2.1

##
## Attaching package: 'fitur'
##
## The following object is masked from 'package:purrr':
##
##     rdunif

```

```
library(visdat)
```

Load data

Import data

```
hm <- read.csv("output_data/1.MICE_cleaned_data.csv")
```

```
# Vectors for selecting genes
```

```
#Lab genes
```

```
# The measurements of IL.12 and IRG6 are done with an other assay and will  
#ignore for now
```

```
Gene_lab <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",  
             "IL1RN", "CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",  
             "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",  
             "TICAM1", "TNF") # "IL.12", "IRG6")
```

```
Genes_wild <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",  
              "IL1RN", "CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",  
              "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",  
              "TICAM1", "TNF") #, "IL.12", "IRG6")
```

```
Facs_lab <- c("CD4", "Treg", "Div_Treg", "Treg17", "Th1",  
            "Div_Th1", "Th17", "Div_Th17", "CD8", "Act_CD8",  
            "Div_Act_CD8", "IFNy_CD4", "IFNy_CD8", "Treg_prop",  
            "IL17A_CD4")
```

```
Facs_wild <- c("Treg", "CD4", "Treg17", "Th1", "Th17", "CD8",  
             "Act_CD8", "IFNy_CD4", "IL17A_CD4", "IFNy_CD8")
```

data imputation

Genes

```
field <- hm %>%  
  dplyr::filter(origin == "Field")  
  
field <- unique(field)  
  
genes_mouse_field <- field %>%  
  dplyr::select(c(Mouse_ID, all_of(Genes_wild)))  
  
genes <- genes_mouse_field %>%
```

```

dplyr::select(-Mouse_ID)

#remove rows with only nas
genes <- genes[,colSums(is.na(genes))<nrow(genes)]

#remove columns with only nas
genes <- genes[rowSums(is.na(genes)) != ncol(genes), ]

genes_mouse_field <- genes_mouse_field[row.names(genes), ]

##select same rows in the first table
field <- field[row.names(genes), ]

#####lab
#select the genes and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid
# duplicates

lab <- unique(lab)

gene_lab_mouse <- lab %>%
  dplyr::select(c(Mouse_ID, all_of(Gene_lab)))

gene_lab_mouse <- unique(gene_lab_mouse)

genes_lab <- gene_lab_mouse[, -1]

#remove rows with only nas
genes_lab <- genes_lab[,colSums(is.na(genes_lab))<nrow(genes_lab)]

#remove columns with only nas
genes_lab <- genes_lab[rowSums(is.na(genes_lab)) != ncol(genes_lab), ]

genes_lab <- unique(genes_lab)

#select same rows in the first table
gene_lab_mouse <- gene_lab_mouse[row.names(genes_lab), ]

##select same rows in the first table
lab <- lab[row.names(genes_lab), ]

hm_genes <- rbind(gene_lab_mouse, genes_mouse_field)

hm_selection_g <- rbind(lab, field)

genes <- hm_genes %>%
  dplyr::select(-Mouse_ID)

```

```
# looking at patterns of nas
#pattern_na <-as.data.frame(md.pattern(field_genes))

sapply(hm_genes, function(x) sum(is.na(x)))
```

```
## Mouse_ID      IFNy      CXCR3      IL.6      IL.13      IL.10      IL1RN      CASP1
##           0         62        110        111        124        230         31        131
##      CXCL9      ID01      IRGM1      MPO       MUC2      MUC5AC      MYD88      NCR1
##          42         29         11         54         14         30         20        139
##      PRF1      RETNLB      SOCS1      TICAM1      TNF
##          158        108         11         121         42
```

```
#had to remove as they were disturbing the imputation: Worms_presence, MC.Eimeria.FEC, Heligmosomoides.
```

```
#vis_miss(field)
```

```
# The frequency distribution of the missing cases per variable can be obtained
# as:
```

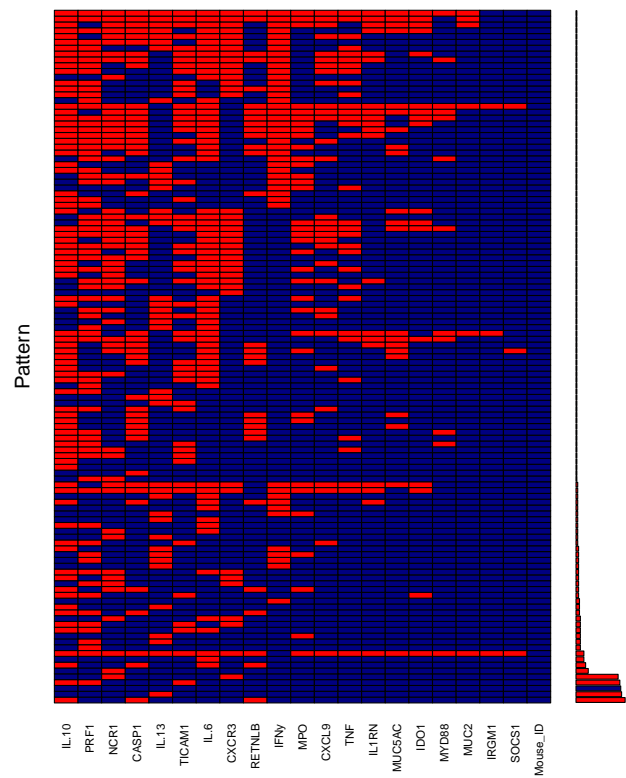
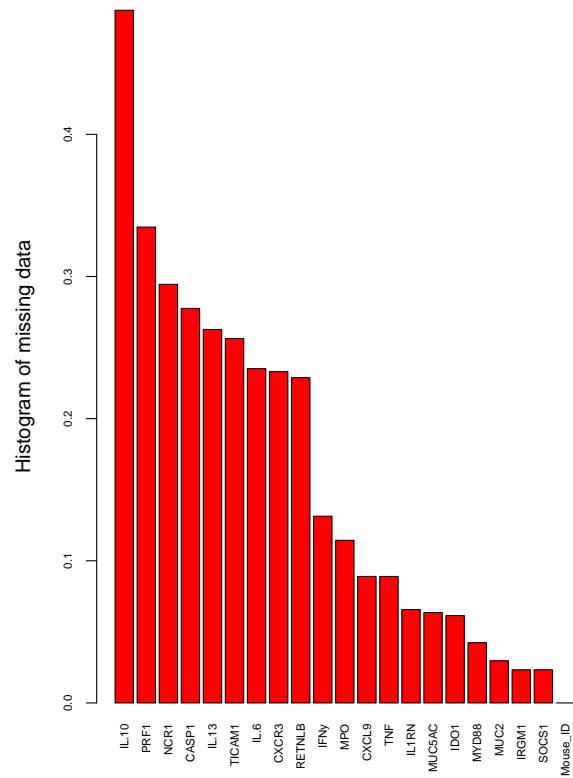
```
init <- mice(genes, maxit = 0)
```

```
#we want to impute only the specific variables
```

```
meth <- init$method
```

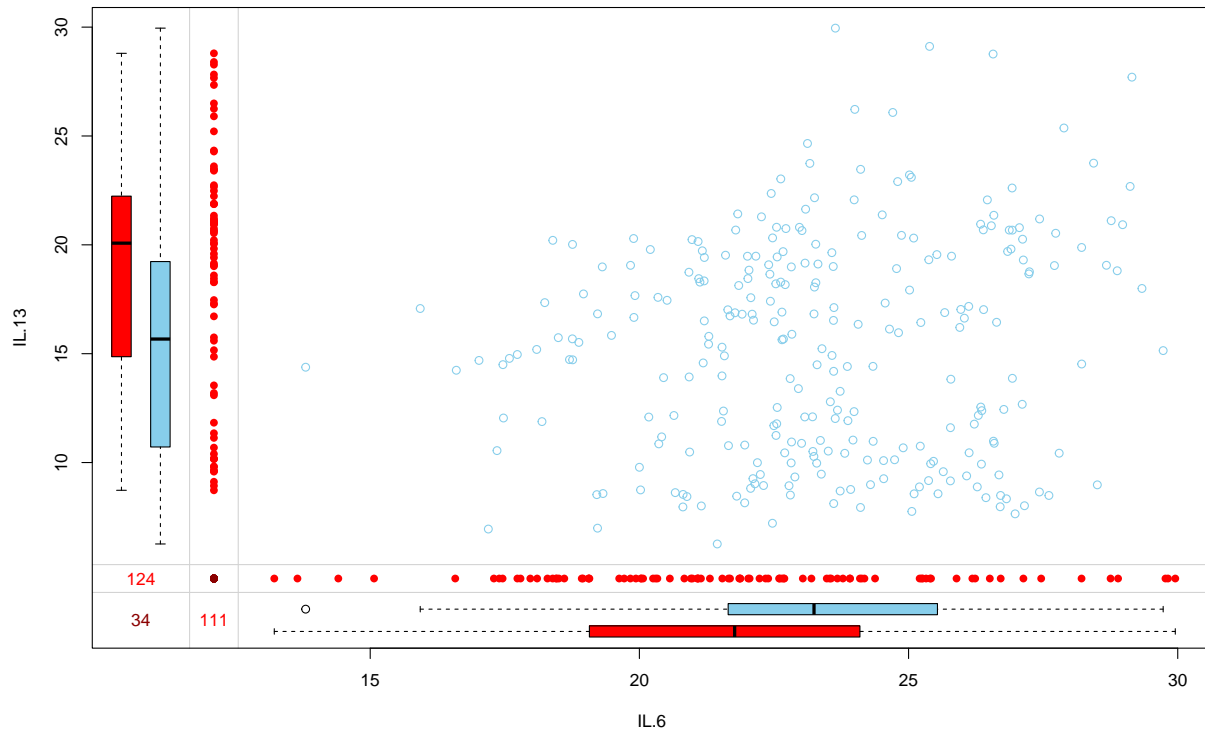
```
aggr_plot <- aggr(hm_genes, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(hm_genes))
```

```
## Warning in plot.aggr(res, ...): not enough vertical space to display frequencies
## (too many combinations)
```



```
##
## Variables sorted by number of missings:
## Variable      Count
## IL.10 0.48728814
## PRF1 0.33474576
## NCR1 0.29449153
## CASP1 0.27754237
## IL.13 0.26271186
## TICAM1 0.25635593
## IL.6 0.23516949
## CXCR3 0.23305085
## RETNLB 0.22881356
## IFNγ 0.13135593
## MPO 0.11440678
## CXCL9 0.08898305
## TNF 0.08898305
## IL1RN 0.06567797
## MUC5AC 0.06355932
## IDO1 0.06144068
## MYD88 0.04237288
## MUC2 0.02966102
## IRGM1 0.02330508
## SOCS1 0.02330508
## Mouse_ID 0.00000000
```

```
marginplot(hm_genes[c(4,5)])
```



```
# m=5 refers to the number of imputed datasets. Five is the default value.
igf <- mice(genes, m = 5, seed = 500) # method = meth,
```

```
##
## iter imp variable
## 1 1 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 1 2 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 1 3 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 1 4 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 1 5 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 2 1 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 2 2 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 2 3 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 2 4 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 2 5 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 3 1 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 3 2 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 3 3 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 3 4 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 3 5 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 4 1 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 4 2 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 4 3 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 4 4 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
```

```
## 4 5 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 5 1 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 5 2 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 5 3 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 5 4 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
## 5 5 IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC MYD88
```

```
summary(igf)
```

```
## Class: mids
## Number of multiple imputations: 5
## Imputation methods:
## IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO
## "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm"
## MUC2 MUC5AC MYD88 NCR1 PRF1 RETNLB SOCS1 TICAM1 TNF
## "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm" "pmm"
## PredictorMatrix:
## IFNy CXCR3 IL.6 IL.13 IL.10 IL1RN CASP1 CXCL9 ID01 IRGM1 MPO MUC2 MUC5AC
## IFNy 0 1 1 1 1 1 1 1 1 1 1 1 1
## CXCR3 1 0 1 1 1 1 1 1 1 1 1 1 1
## IL.6 1 1 0 1 1 1 1 1 1 1 1 1 1
## IL.13 1 1 1 0 1 1 1 1 1 1 1 1 1
## IL.10 1 1 1 1 0 1 1 1 1 1 1 1 1
## IL1RN 1 1 1 1 1 0 1 1 1 1 1 1 1
## MYD88 NCR1 PRF1 RETNLB SOCS1 TICAM1 TNF
## IFNy 1 1 1 1 1 1 1
## CXCR3 1 1 1 1 1 1 1
## IL.6 1 1 1 1 1 1 1
## IL.13 1 1 1 1 1 1 1
## IL.10 1 1 1 1 1 1 1
## IL1RN 1 1 1 1 1 1 1
```

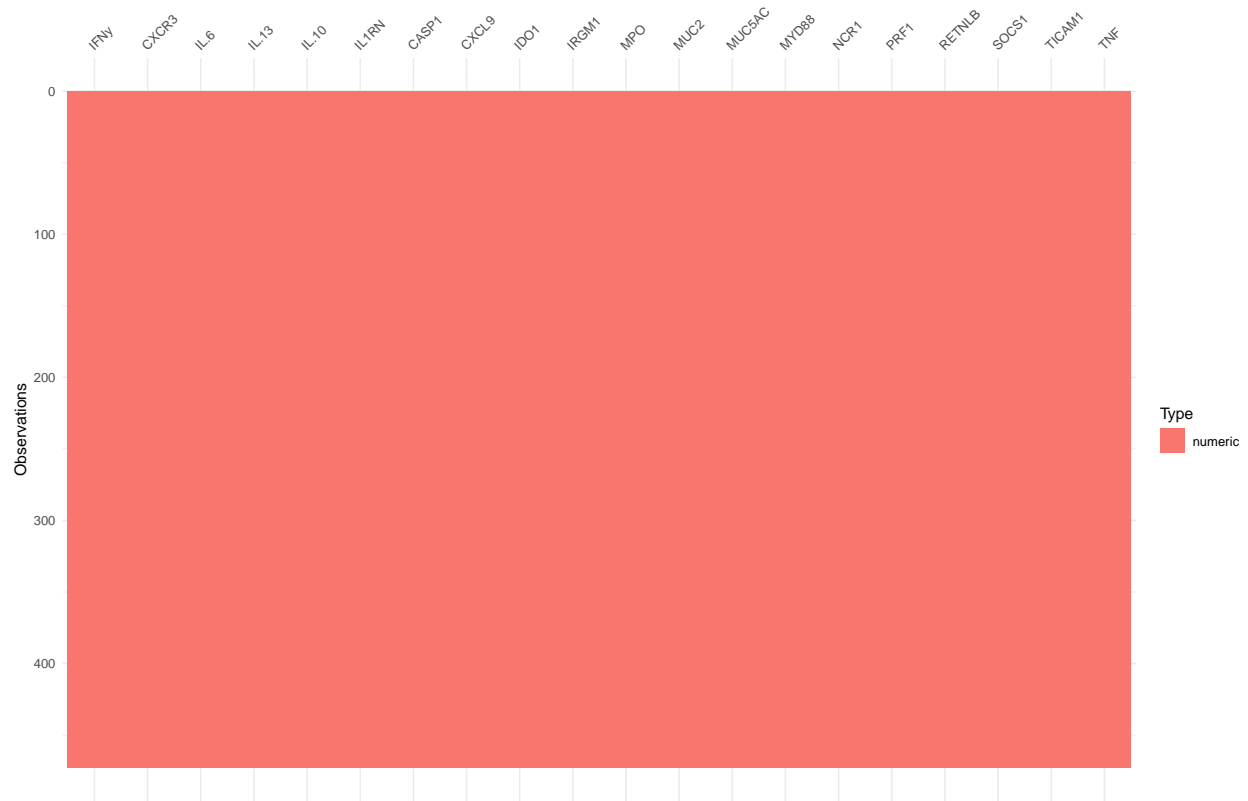
```
# to check each column with imputed data
## igf$imp$IFNy

#Now we can get back the completed dataset using the complete()
complete_genes <- complete(igf, 1)

#sapply(complete_field, function(x) sum(is.na(x)))

#visualize missingness
vis_dat(complete_genes)
```

```
## Warning: 'gather_()' was deprecated in tidyr 1.2.0.
## i Please use 'gather()' instead.
## i The deprecated feature was likely used in the visdat package.
## Please report the issue at <https://github.com/ropensci/visdat/issues>.
```

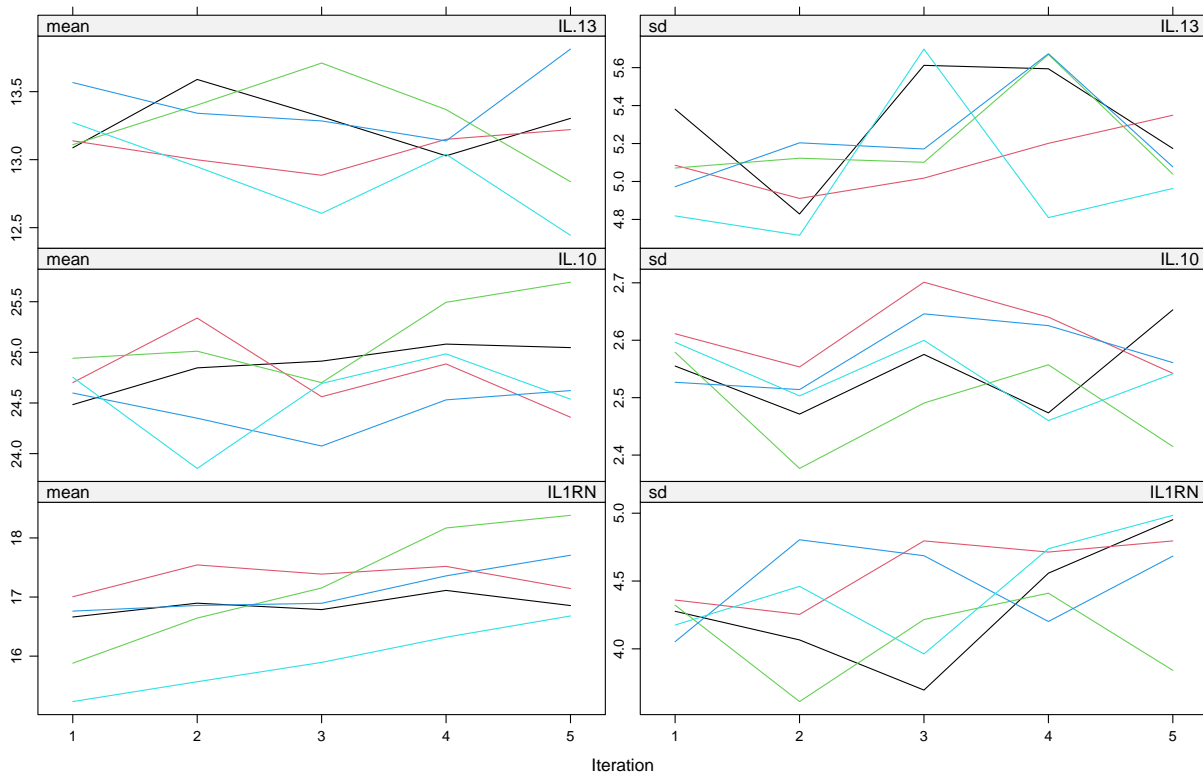
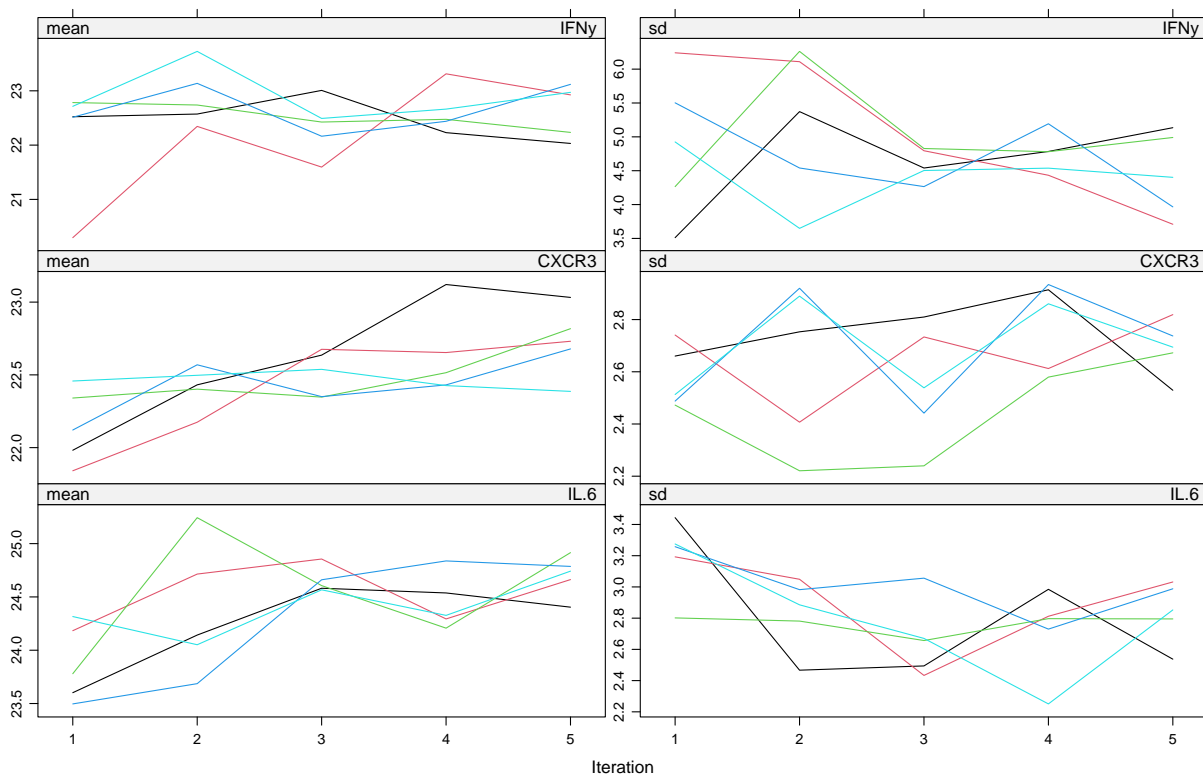



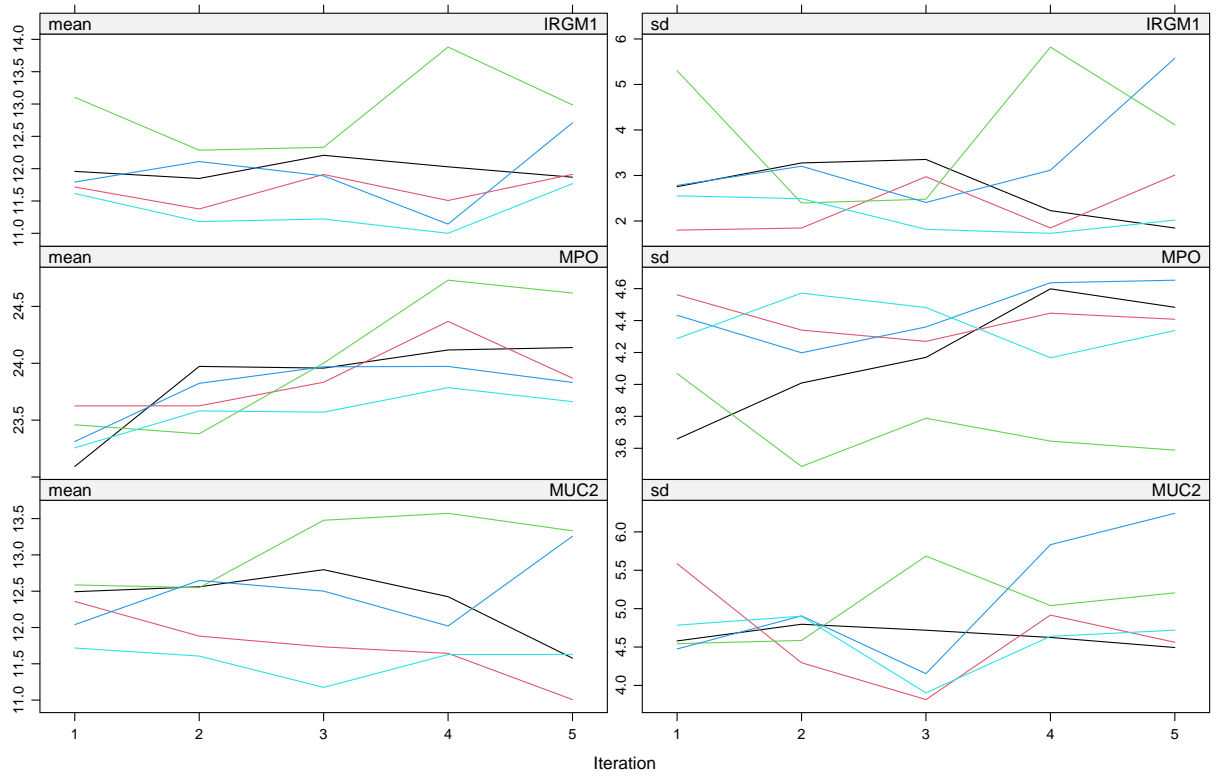
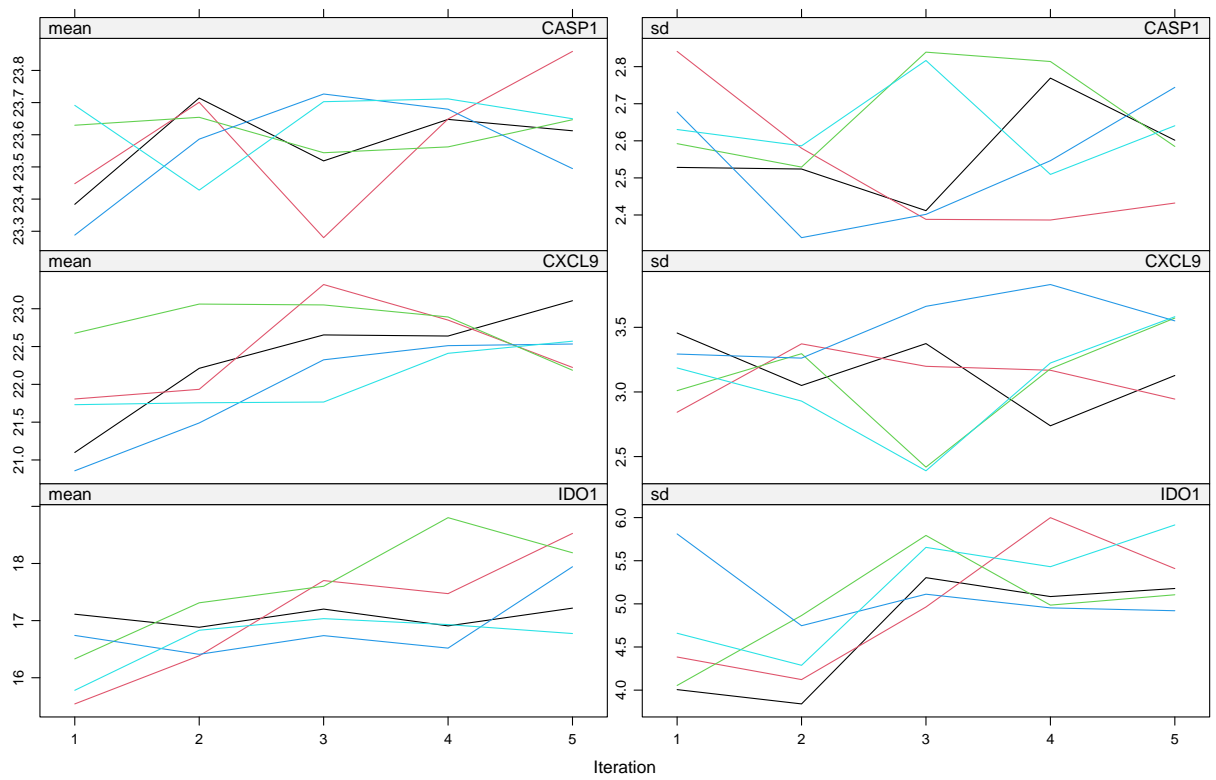
```
#remove the non imputed genes from our data set
hm_selection_g <- hm_selection_g %>%
  dplyr::select(-c("IFNγ", "CXCR3", "IL.6", "IL.13", "IL.10",
                  "IL1RN", "CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF"))

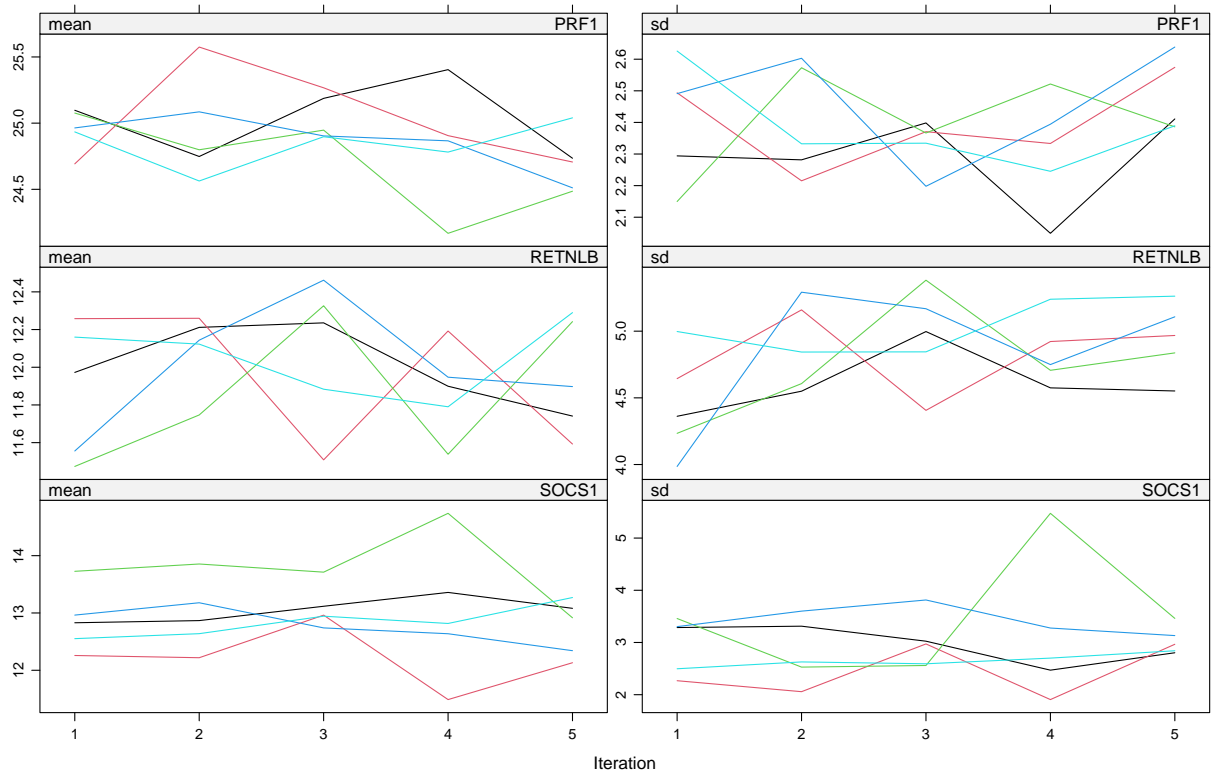
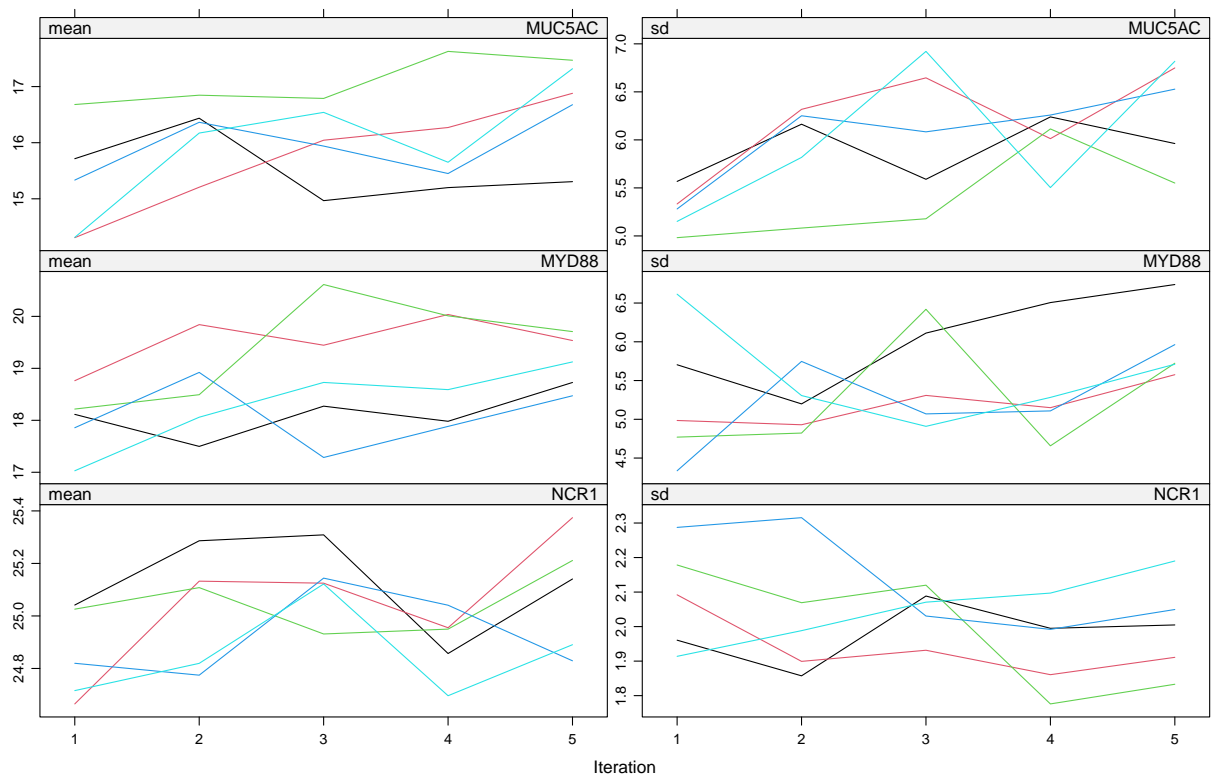
# add the new imputed genes to the data
hm_selection_g <- cbind(hm_selection_g, complete_genes)
```

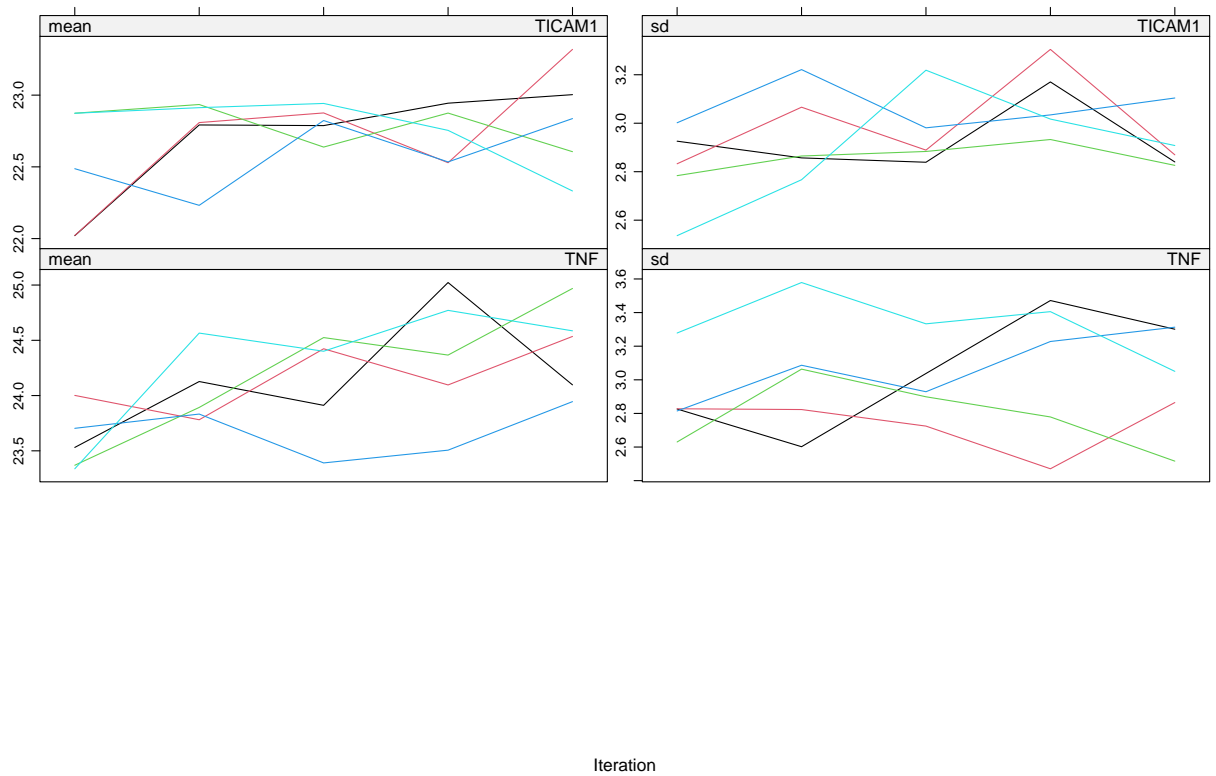
inspect the trace lines for convergence:

```
plot(igf)
```



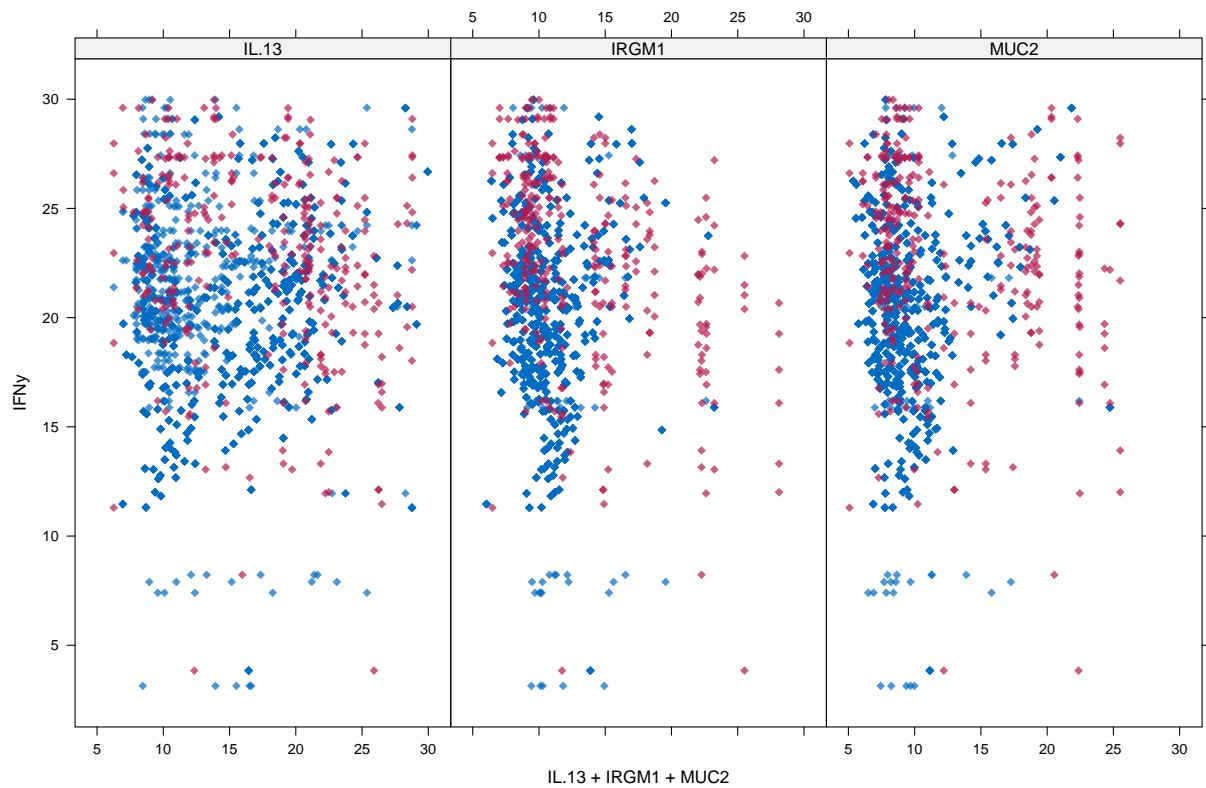






Let's compare the distributions of original and imputed data using a some useful plots. First of all we can use a scatterplot and plot Ozone against all the other variables. Let's first plot the variables for which we have few missing values.

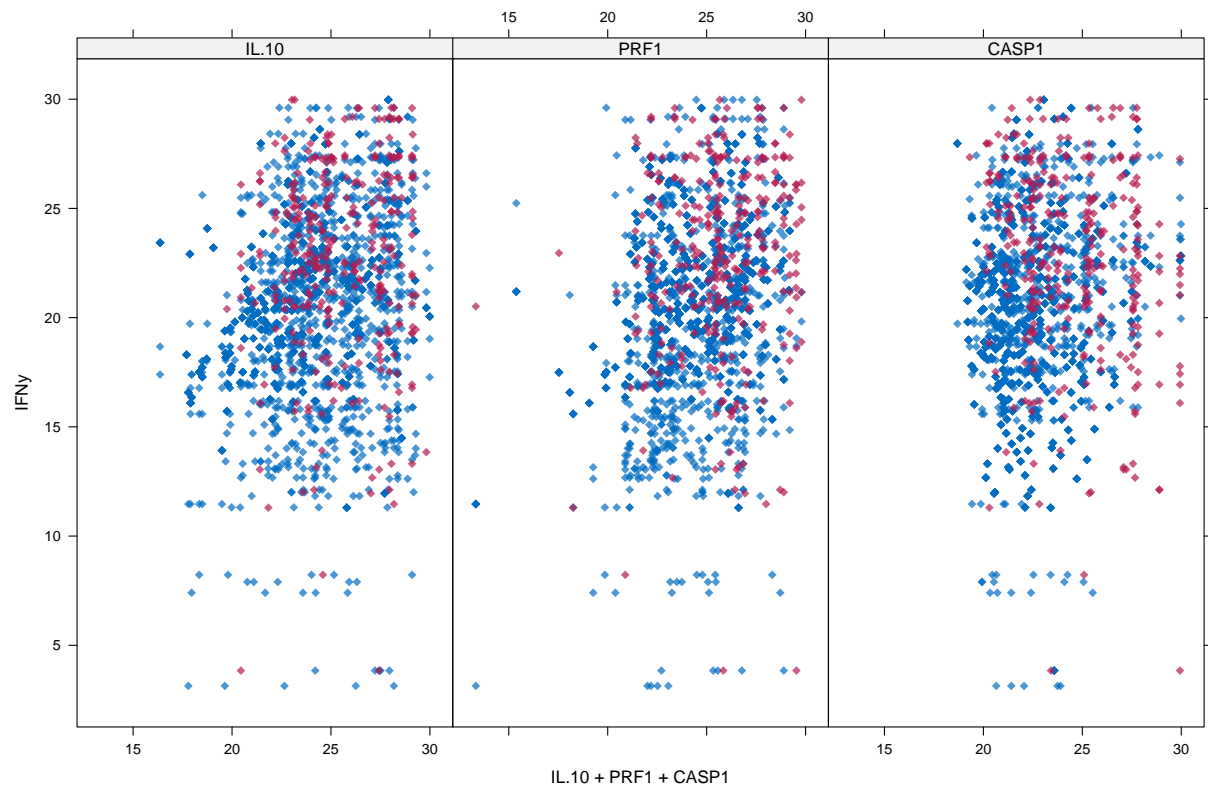
```
xyplot(igf, IFNy ~ IL.13 + IRGM1 + MUC2, pch=18, cex=1)
```



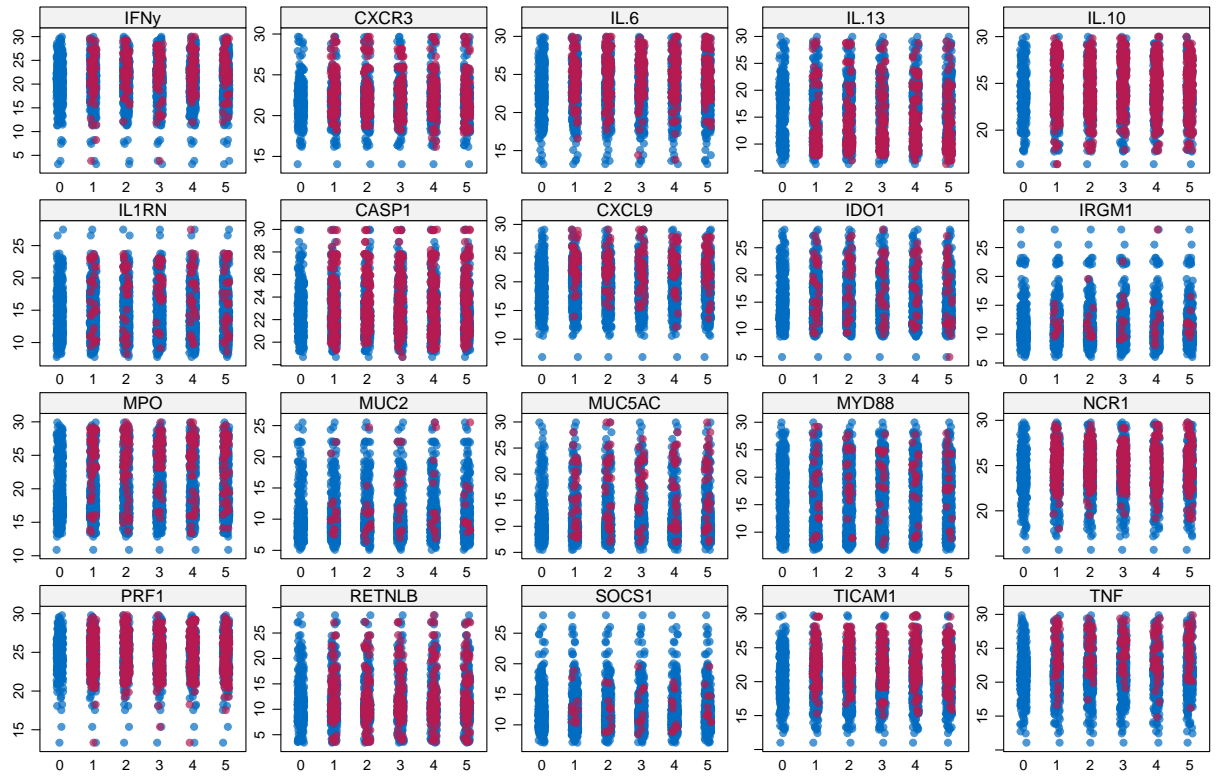
What we would like to see is that the shape of the magenta points (imputed) matches the shape of the blue ones (observed). The matching shape tells us that the imputed values are indeed “plausible values”.

Now let’s plot the variables with many missing data points.

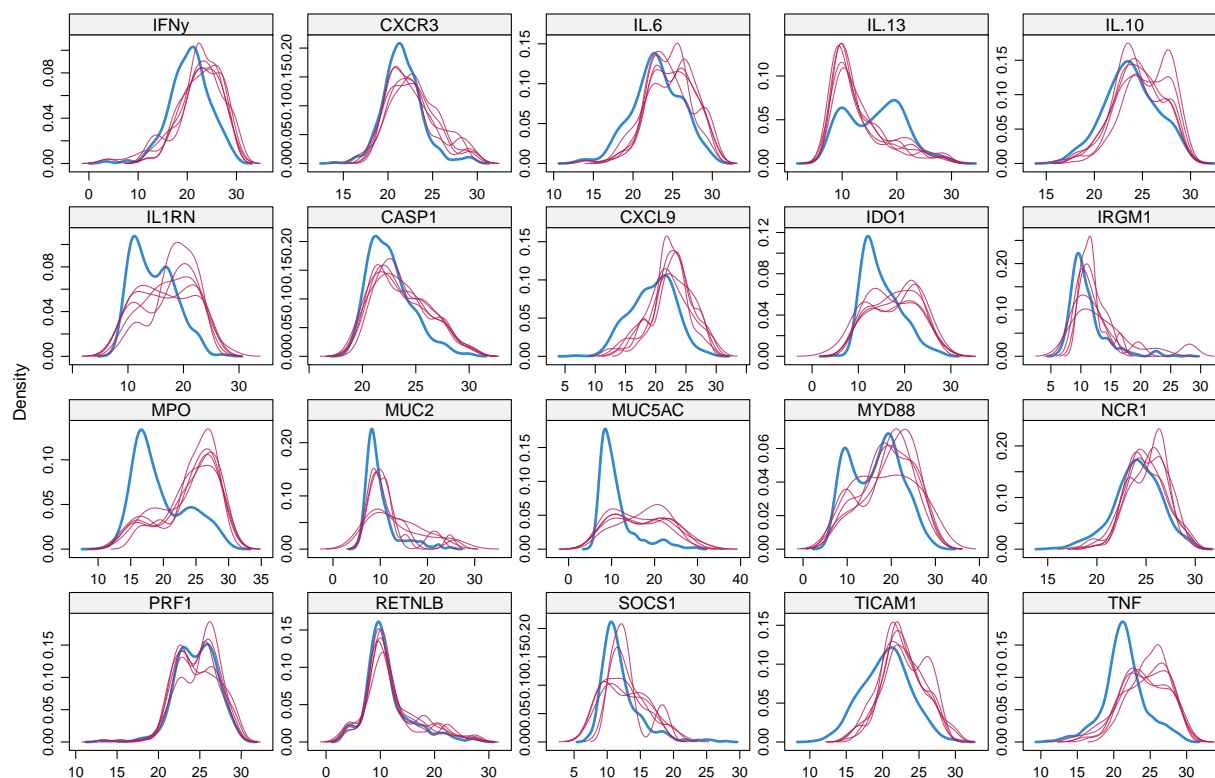
```
xyplot(igf, IFNy ~ IL.10 + PRF1 + CASP1, pch=18, cex=1)
```



```
stripplot(igf, pch = 20, cex = 1.2)
```



```
densityplot(igf)
```

The density of the imputed data for each imputed dataset is showed in magenta while the density of the observed data is showed in blue. Again, under our previous assumptions we expect the distributions to be similar.

Another useful visual take on the distributions can be obtained using the `stripplot()` function that shows the distributions of the variables as individual points

Facs

```
#####lab
#select the facs and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid
# duplicates

lab <- unique(lab)

facs_mouse <- lab %>%
  dplyr::select(c(Mouse_ID, all_of(Facs_lab))) #choosing the same with the wild

facs_mouse <- unique(facs_mouse)

facs_lab <- facs_mouse[, -1]

#remove rows with only nas
```

```

facs_lab <- facs_lab[,colSums(is.na(facs_lab))<nrow(facs_lab)]

#remove columns with only nas
facs_lab <- facs_lab[rowSums(is.na(facs_lab)) != ncol(facs_lab), ]

#select same rows in the first table
facs_mouse <- facs_mouse[row.names(facs_lab), ]

##select same rows in the first table
lab <- lab[row.names(facs_mouse), ]

#####Field
#####field
# somehow the field samples have the origin na,
# fix that

field <- hm %>%
  dplyr::filter(origin == "Field")

field <- unique(field)

facs_mouse <- field %>%
  dplyr::select(c(Mouse_ID, all_of(Facs_wild)))

facs_field <- facs_mouse[,-1]

#remove rows with only nas
facs_field <- facs_field[,colSums(is.na(facs_field))<nrow(facs_field)]

#remove columns with only nas
facs_field <- facs_field[rowSums(is.na(facs_field)) != ncol(facs_field), ]

##select same rows in the first table
field <- field[row.names(facs_field), ]

facs_data <- full_join(lab, field, by = intersect(colnames(lab), colnames(field)))

facs_data <- unique(facs_data) %>%
  dplyr::select(-c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",
                  "IL1RN", "CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF"))

```

Now put everything together

```

hm_selection_g <- hm_selection_g %>%
  full_join(facs_data, by = intersect(colnames(facs_data), colnames(hm_selection_g)))

```

```
hm_selection_g <- unique(hm_selection_g)
```

```
##save the imputed data  
write.csv(hm_selection_g, "output_data/2.imputed_MICE_data_set.csv", row.names = FALSE)
```