Hands-On Part

The following real data example is adapted to a large extend from the guidance on Rmisstastic_descriptive_statistics_with_missing_values

Air Quality Data

Air pollution is currently one of the most serious public health worries worldwide. Many epidemiological studies have proved the influence that some chemical compounds, such as sulphur dioxide (SO_2), nitrogen dioxide (NO_2), ozone (O_3) can have on our health. Associations set up to monitor air quality are active all over the world to measure the concentration of these pollutants.

The data set we use here is a small subset of a cleaned version of air pollution measurements in the US. For more details, I refer to the Appendix C of the following paper. In this example, I actually induced missing values here, so that we have full control over the missing mechanism and access to the true data.

We first load a real (prepared) data set:

```
library(mice)

# Naniar provides principled, tidy ways to summarise, visualise, and manipulate
# missing data with minimal deviations from the workflows in ggplot2 and tidy
# data.
library(naniar)
library(VIM)
library(FactoMineR)

X <- read.csv("data.csv", header = T, row.names = 1)
Xstar<- read.csv("fulldata.csv", header = T, row.names = 1)

head(X)
```

```
##   max_PM2.5 max_O3 max_PM10  Longitude Latitude Elevation Land.Use_AGRICULTURAL
## 1      4.75  0.027       11 -106.58520 35.13430      1591                     0
## 2      7.15  0.055       25 -120.68028 38.20185        NA                     1
## 3      1.45  0.044        8 -105.30353 42.76697      1508                     1
## 4      7.00  0.058       20  -70.74802 43.07537         8                     0
## 5     12.00  0.017       NA -112.09577 33.50383        NA                     0
## 6      5.40  0.049       17  -85.89804 38.06091       135                     0
##   Land.Use_COMMERCIAL Land.Use_INDUSTRIAL Location.Setting_RURAL
## 1                   0                   0                      0
## 2                   0                   0                      1
## 3                   0                   0                      1
## 4                   0                   0                      0
## 5                   0                   0                      0
## 6                   0                   0                      0
##   Location.Setting_SUBURBAN
```

```
## 1                          0
## 2                          0
## 3                          0
## 4                          0
## 5                          0
## 6                          1
```

```r
head(Xstar)
```

```
##   max_PM2.5 max_O3 max_PM10  Longitude Latitude Elevation Land.Use_AGRICULTURAL
## 1      4.75  0.027       11 -106.58520 35.13430      1591                     0
## 2      7.15  0.055       25 -120.68028 38.20185       310                     1
## 3      1.45  0.044        8 -105.30353 42.76697      1508                     1
## 4      7.00  0.058       20  -70.74802 43.07537         8                     0
## 5     12.00  0.017       15 -112.09577 33.50383       343                     0
## 6      5.40  0.049       17  -85.89804 38.06091       135                     0
##   Land.Use_COMMERCIAL Land.Use_INDUSTRIAL Location.Setting_RURAL
## 1                   0                   0                      0
## 2                   0                   0                      1
## 3                   0                   0                      1
## 4                   0                   0                      0
## 5                   0                   0                      0
## 6                   0                   0                      0
##   Location.Setting_SUBURBAN
## 1                         0
## 2                         0
## 3                         0
## 4                         0
## 5                         0
## 6                         1
```

```r
summary(X)
```

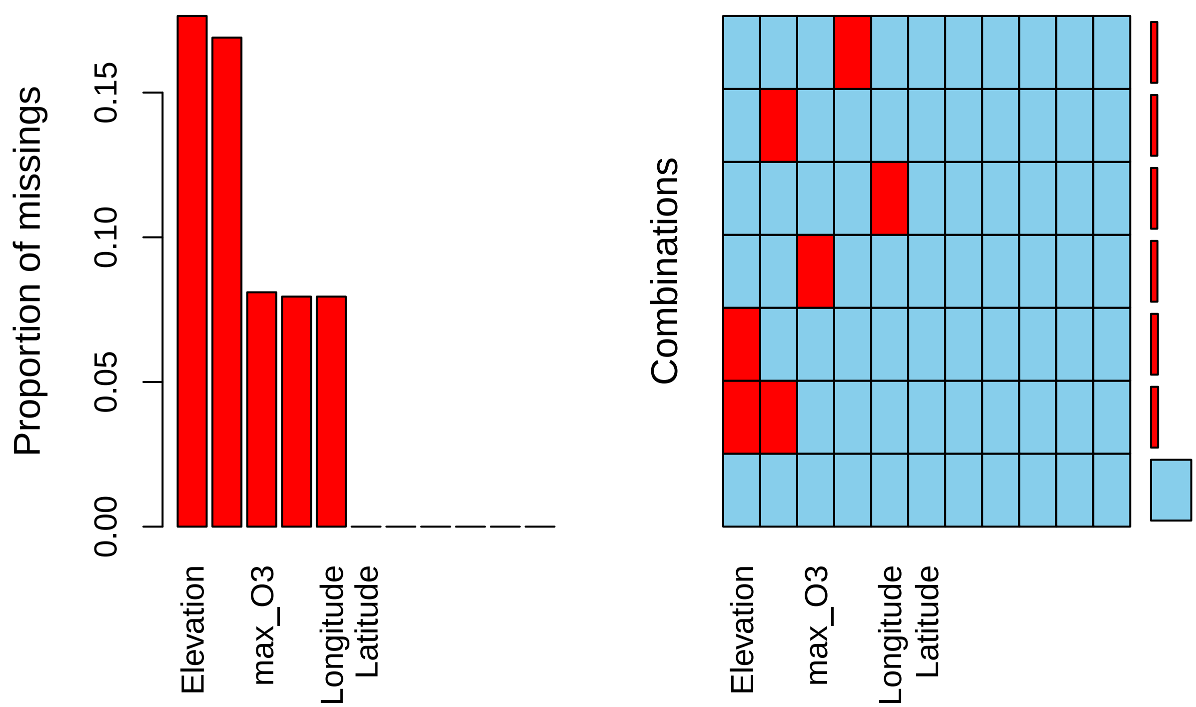
```
##    max_PM2.5          max_O3          max_PM10        Longitude
##  Min.   :-3.300   Min.   :0.0010   Min.   :  0.00   Min.   :-124.18
##  1st Qu.: 4.100   1st Qu.:0.0330   1st Qu.:  9.00   1st Qu.:-117.33
##  Median : 6.200   Median :0.0410   Median : 15.00   Median :-106.51
##  Mean   : 7.332   Mean   :0.0413   Mean   : 18.31   Mean   :-102.33
##  3rd Qu.: 9.200   3rd Qu.:0.0490   3rd Qu.: 23.00   3rd Qu.: -88.22
##  Max.   :56.333   Max.   :0.0910   Max.   :143.00   Max.   : -68.26
##  NA's   :159      NA's   :162      NA's   :338      NA's   :159
##     Latitude       Elevation    Land.Use_AGRICULTURAL Land.Use_COMMERCIAL
##  Min.   :26.05   Min.   : -14   Min.   :0.000         Min.   :0.0000
##  1st Qu.:34.49   1st Qu.:  68   1st Qu.:0.000         1st Qu.:0.0000
##  Median :38.20   Median : 269   Median :0.000         Median :0.0000
##  Mean   :38.35   Mean   : 430   Mean   :0.125         Mean   :0.3115
##  3rd Qu.:41.30   3rd Qu.: 571   3rd Qu.:0.000         3rd Qu.:1.0000
##  Max.   :48.64   Max.   :2195   Max.   :1.000         Max.   :1.0000
##                  NA's   :353
##  Land.Use_INDUSTRIAL Location.Setting_RURAL Location.Setting_SUBURBAN
##  Min.   :0.00        Min.   :0.000          Min.   :0.000
##  1st Qu.:0.00        1st Qu.:0.000          1st Qu.:0.000
##  Median :0.00        Median :0.000          Median :0.000
```

```
##  Mean   :0.03        Mean   :0.193          Mean   :0.428
##  3rd Qu.:0.00        3rd Qu.:0.000          3rd Qu.:1.000
##  Max.   :1.00        Max.   :1.000          Max.   :1.000
##
```

1) Descriptive statistics with missing values

We start by inspecting various plots for the missing values:

```
res <- summary(aggr(X, sortVar = TRUE))$combinations
```



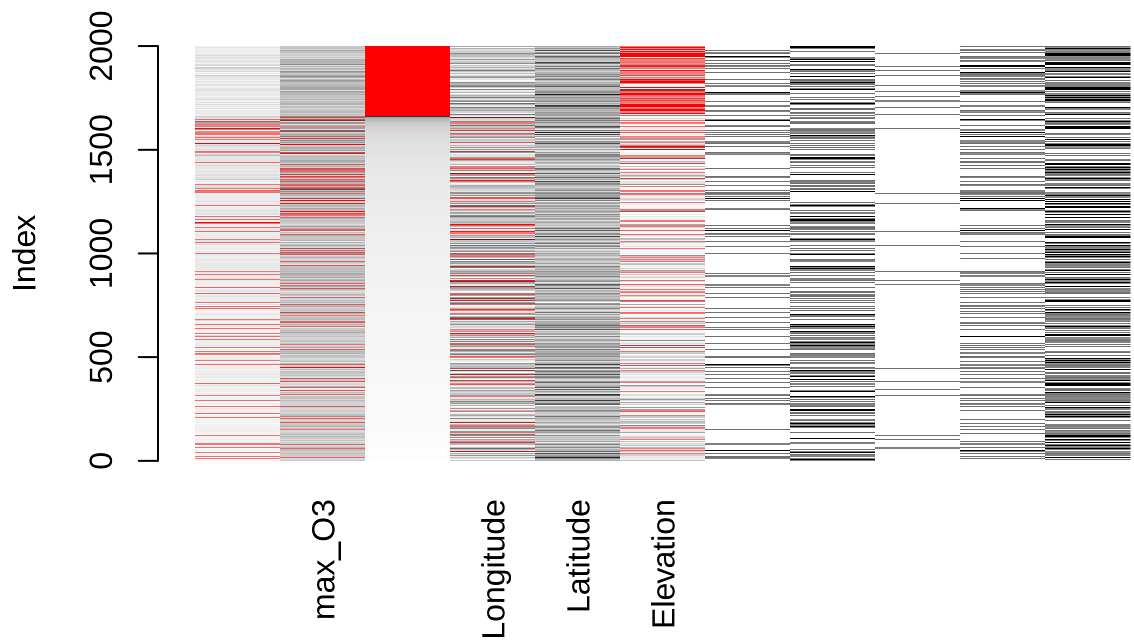

```
##
##  Variables sorted by number of missings:
##                   Variable  Count
##                  Elevation 0.1765
##                   max_PM10 0.1690
##                     max_O3 0.0810
##                  max_PM2.5 0.0795
##                  Longitude 0.0795
##                   Latitude 0.0000
##      Land.Use_AGRICULTURAL 0.0000
##        Land.Use_COMMERCIAL 0.0000
##        Land.Use_INDUSTRIAL 0.0000
##     Location.Setting_RURAL 0.0000
##  Location.Setting_SUBURBAN 0.0000
```

```r
res[rev(order(res[, 2])), ]
```

```
##            Combinations Count Percent
## 1 0:0:0:0:0:0:0:0:0:0:0  1008   50.40
## 5 0:0:1:0:0:1:0:0:0:0:0   179    8.95
## 2 0:0:0:0:0:1:0:0:0:0:0   174    8.70
## 6 0:1:0:0:0:0:0:0:0:0:0   162    8.10
## 7 1:0:0:0:0:0:0:0:0:0:0   159    7.95
## 4 0:0:1:0:0:0:0:0:0:0:0   159    7.95
## 3 0:0:0:1:0:0:0:0:0:0:0   159    7.95
```

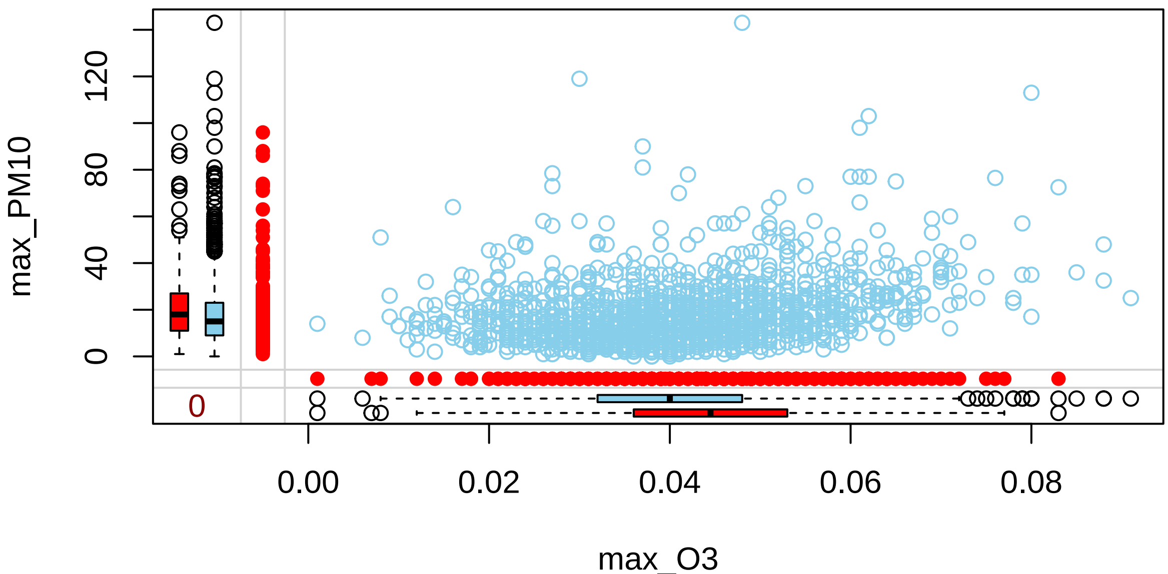
Creating the res variable renders a nice plot, showing the percentage of missing values for each variable. Moreover the next command nicely shows the patterns (M), as well as their frequency of occurring in the data set. In particular, we can further visualize the pattern using the matrixplot function:

```r
matrixplot(X, sortby = 3)
```



The **VIM** function `marginplot` creates a scatterplot with additional information on the missing values. If you plot the variables (x, y), the points with no missing values are represented as in a standard scatterplot. The points for which x (resp. y) is missing are represented in red along the y (resp. x) axis. In addition, boxplots of the x and y variables are represented along the axes with and without missing values (in red all variables x where y is missing, in blue all variables x where y is observed).

```r
marginplot(X[,2:3])
```



This plot can be used to check whether MCAR might hold. Under MCAR, the distribution of a variable when another variable is missing should always be the same. Under MAR this can be violated as we have seen (distribution shifts!). This plotting is a convenient way to check this a bit.

There are many more plotting possibilities with VIM, as demonstrated e.g., in 2012ADAC.pdf (tuwien.ac.at).

2) Imputation

We now finally use the **mice** package for imputation.

```
library(mice)
```

We consider several methods and then start by choosing the best one according to the new I-Score. I-Score is contained in **miceDRF** package. In can be installed with

```
devtools::install_github("KrystynaGrzesiak/miceDRF")
```

As the best version of the score not only scores one imputation but an imputation method itself for this dataset, we need to define a function for each:

```
library(miceDRF)

X <- as.matrix(X)

methods <- c("pmm",       # mice-pmm
             "cart",      # mice-cart
             "sample",    # mice-sample
```

```r
             "norm.nob", # Gaussian Imputation
             "DRF")      # mice-DRF


# Creating a list of impuattion functions
imputation_list <- create_mice_imputations(methods)

# Calculatiung the scores
scores <- Iscores_compare(X = X, N = 30,
                          imputation_list = imputation_list,
                          methods = methods)

scores
```

The score considers mice-cart to to be the best method. As a side note however, mice-rf is deemed second best and might have better properties for uncertainty estimation and multiple imputation, thus both should be considered. Here, we go with mice-cart:

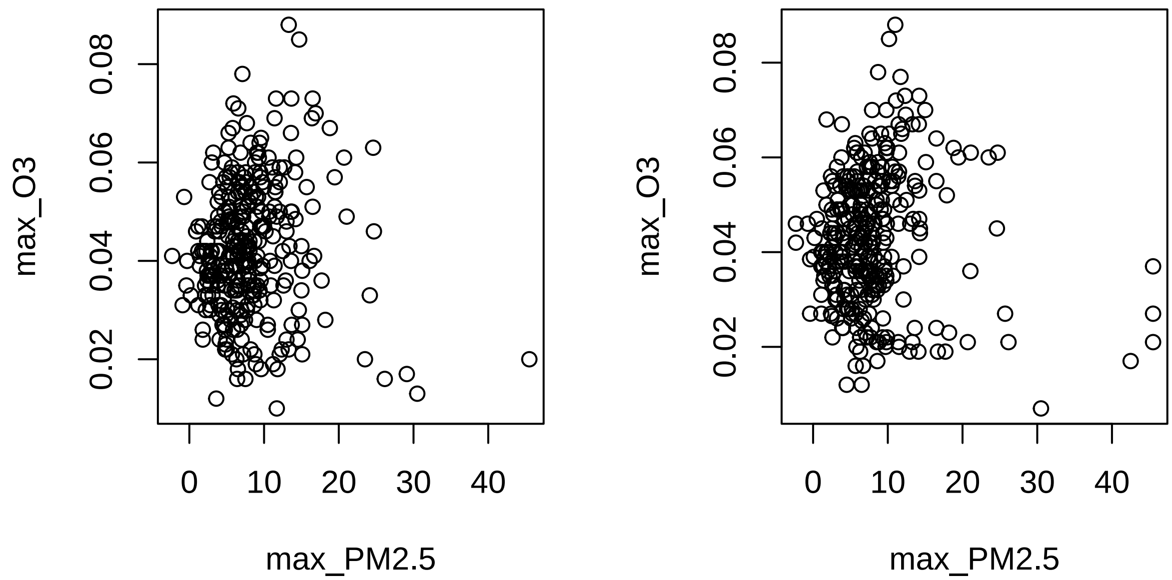
```r
imp.mice <- mice(X, m = 10, method = "cart", printFlag = F)
```

Since we have the true data in this case, we analyze the imputation method a bit closer:

```r
## This here is not possible without the fully observed data ###
Ximp <- mice::complete(imp.mice)

index1 <- 1
index2 <- 2

par(mfrow = c(1, 2))
plot(Xstar[is.na(X[, index1]) | is.na(X[, index2]), c(index1, index2)])
plot(Ximp[is.na(X[, index1]) | is.na(X[, index2]), c(index1, index2)])
```




```r
# Replicating first and second moments
colMeans(Xstar) - colMeans(Ximp)
```

```
##                    max_PM2.5                      max_O3                    max_PM10
##                  0.010471667                -0.000034250                -0.062416667
##                    Longitude                    Latitude                   Elevation
##                 -0.009294513                 0.000000000                -0.362557202
##       Land.Use_AGRICULTURAL         Land.Use_COMMERCIAL         Land.Use_INDUSTRIAL
##                  0.000000000                 0.000000000                 0.000000000
##     Location.Setting_RURAL Location.Setting_SUBURBAN
##                  0.000000000                 0.000000000
```

```r
norm(cov(Xstar) - cov(Ximp)) / norm(cov(Xstar))
```

```
## [1] 0.00321098
```

3) Analyse

```r
# Apply a regression to the multiple imputation
lm.mice.out <- with(imp.mice, lm(max_O3 ~ max_PM2.5 + Longitude + Latitude + Elevation + Land.Use_AGRICU

# Use Rubins Rules to aggregate the estimates
res <- pool(lm.mice.out)
summary(res)
```

```
##                          term      estimate    std.error   statistic        df
## 1                (Intercept)  3.921278e-02 3.715559e-03 10.55366814 1342.0968
## 2                  max_PM2.5  1.252037e-04 5.949398e-05  2.10447672  149.4103
## 3                  Longitude -1.236763e-05 2.037901e-05 -0.60688084 1351.4156
## 4                   Latitude -9.482421e-06 7.507446e-05 -0.12630688  671.6427
## 5                  Elevation -1.142371e-06 7.410295e-07 -1.54160021  289.8863
## 6      Land.Use_AGRICULTURAL -3.066870e-05 1.450127e-03 -0.02114897  698.3616
## 7        Land.Use_COMMERCIAL  6.564170e-04 7.004739e-04  0.93710424 1160.7476
## 8        Land.Use_INDUSTRIAL -1.683155e-03 1.832320e-03 -0.91859271  774.3648
## 9     Location.Setting_RURAL  1.783081e-03 1.170311e-03  1.52359593 1671.3800
## 10 Location.Setting_SUBURBAN  7.651281e-04 6.851502e-04  1.11673049 1095.4053
##         p.value
## 1  4.543801e-25
## 2  3.701099e-02
## 3  5.440319e-01
## 4  8.995268e-01
## 5  1.242616e-01
## 6  9.831329e-01
## 7  3.488999e-01
## 8  3.585947e-01
## 9  1.277989e-01
## 10 2.643545e-01
```

Importantly, this works here because we have all the ingredients for the `pool` function, which are (according to `?pool`):

- the estimates of the model;
- the standard error of each estimate;
- the residual degrees of freedom of the model.

Just to double check, we also perform the regression on X^*:

```
## This here is not possible without the fully observed data ###
res.not.attainable <- lm(max_O3 ~ max_PM2.5 + Longitude + Latitude + Elevation + Land.Use_AGRICULTURAL +

summary(res.not.attainable)
```

```
##
## Call:
## lm(formula = max_O3 ~ max_PM2.5 + Longitude + Latitude + Elevation +
##     Land.Use_AGRICULTURAL + Land.Use_COMMERCIAL + Land.Use_INDUSTRIAL +
##     Location.Setting_RURAL + Location.Setting_SUBURBAN, data = as.data.frame(Xstar))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043412 -0.008181 -0.000635  0.007901  0.049148
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.092e-02  3.627e-03  11.282   <2e-16 ***
## max_PM2.5                  1.305e-04  5.333e-05   2.447   0.0145 *
```

```
## Longitude                   1.314e-08  1.988e-05   0.001   0.9995
## Latitude                   -2.495e-05  7.142e-05  -0.349   0.7269
## Elevation                  -9.333e-07  6.769e-07  -1.379   0.1681
## Land.Use_AGRICULTURAL       3.083e-04  1.382e-03   0.223   0.8235
## Land.Use_COMMERCIAL         5.986e-04  6.794e-04   0.881   0.3784
## Land.Use_INDUSTRIAL        -1.777e-03  1.751e-03  -1.015   0.3103
## Location.Setting_RURAL      1.883e-03  1.152e-03   1.634   0.1023
## Location.Setting_SUBURBAN  8.635e-04  6.632e-04   1.302   0.1931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01288 on 1990 degrees of freedom
## Multiple R-squared:  0.007379,   Adjusted R-squared:  0.00289
## F-statistic: 1.644 on 9 and 1990 DF,  p-value: 0.09762
```

```r
cbind(round((res$pooled$estimate - res.not.attainable$coefficients) / res.not.attainable$coefficients, 3
```

```
##                                [,1]
## (Intercept)                  -0.042
## max_PM2.5                    -0.040
## Longitude                  -941.874
## Latitude                     -0.620
## Elevation                     0.224
## Land.Use_AGRICULTURAL        -1.099
## Land.Use_COMMERCIAL           0.097
## Land.Use_INDUSTRIAL          -0.053
## Location.Setting_RURAL       -0.053
## Location.Setting_SUBURBAN    -0.114
```

Of course there are many more challenges, especially also for data that may be partly dependent (for instance repeat measurement or panel data). Most importantly, mice-cart is awesome, but it does not model the uncertainty of the missing imputation itself. As such it is technically not a proper imputation method, as one part of the uncertainty is missing. This could be an issue for confidence intervals and p-values especially in smaller samples. We also refer to the provided links for more information. In particular also the task view on missing data.