Earthquake_MissingValues_Project

Davide Zicca

21/04/2021

Motivazione e caricamento dataset

Obiettivo del seguente progetto è quello di mostrare l'eventuale presenza di Missing Value all'interno di un dataset. Il dataset proposto (fonte: https://www.kaggle.com/srijya/us-earthquake-intensity-database) contiene una collezione di dati di più di 23,000 Terremoti avvenuti negli USA. I dati raccolti presentano gli anni dal 1638 al 1985. Include anche informazioni su *epicentral coordinates, magnitudes, focal depths, names and coordinates of reporting cities (or localities), reported intensities, and the distance from the city (or locality) to the epicenter*. Contiene informazioni anche di altri Stati come *Antigua and Barbuda, Canada, Mexico, Panama, and the Philippines*.

Procedo al caricamento del dataset e alla selezione di alcune delle tante variabili presenti nello stesso:

```
# SOURCE: https://www.kaggle.com/srijya/us-earthquake-intensity-database
# setwd("C:/Davide/MASTER IN DATA SCIENCE/Materiale del Master/Missing Value/PROGETTO")
library(readxl)
eqint_tsqp <- read_excel("eqint_tsqp.xlsx",
                         sheet = "HAZ.EQINT_TSQP")
df= eqint_tsqp
# Missing values

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
# Seleziono le variabili di interesse
df_long <- df%>% select (YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, LATITUDE,
                             LONGITUDE, MAGNITUDE, EQ_DEPTH,EPIDIST, CITY_LAT,
                             CITY_LON, STATE, CITY)
# seleziono 500 indici casuali
rand_ind <- sample (1: nrow (df_long), 500)
df_1 <- df_long [rand_ind,]
```

Verifica presenza Missing Value

```
# install.packages ("naniar")
library(naniar)
```

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

```
# Ci sono valori mancanti nel set di dati?
any_na (df_1)
```

```
## [1] TRUE
```

```
# Quanti?
n_miss (df_1)
```

```
## [1] 816
```

```
prop_miss (df_1) # proportion of missing values
```

```
## [1] 0.1088
```

```
# Quali variabili sono interessate?
df_1%>% is.na ()%>% colSums ()
```

```
##      YEAR     MONTH       DAY      HOUR    MINUTE    SECOND  LATITUDE LONGITUDE
##         0         0         0         3         5       121        62        62
## MAGNITUDE  EQ_DEPTH   EPIDIST  CITY_LAT  CITY_LON     STATE      CITY
##       163       327        65         3         3         2         0
```

```
# Ottieni il numero di missing per variabile (ne%)
miss_var_summary (df_1)
```

```
## # A tibble: 15 x 3
##    variable  n_miss pct_miss
##    <chr>      <int>    <dbl>
##  1 EQ_DEPTH     327     65.4
##  2 MAGNITUDE    163     32.6
##  3 SECOND       121     24.2
##  4 EPIDIST       65     13
##  5 LATITUDE      62     12.4
##  6 LONGITUDE     62     12.4
##  7 MINUTE         5      1
##  8 HOUR           3      0.6
##  9 CITY_LAT       3      0.6
## 10 CITY_LON       3      0.6
## 11 STATE          2      0.4
## 12 YEAR           0      0
## 13 MONTH          0      0
## 14 DAY            0      0
## 15 CITY           0      0
```

```
miss_var_table (df_1)
```

```
## # A tibble: 9 x 3
##   n_miss_in_var n_vars pct_vars
## *         <int>  <int>    <dbl>
## 1             0      4    26.7
## 2             2      1     6.67
## 3             3      3    20
```

```
## 4             5      1     6.67
## 5            62      2    13.3
## 6            65      1     6.67
## 7           121      1     6.67
## 8           163      1     6.67
## 9           327      1     6.67
```

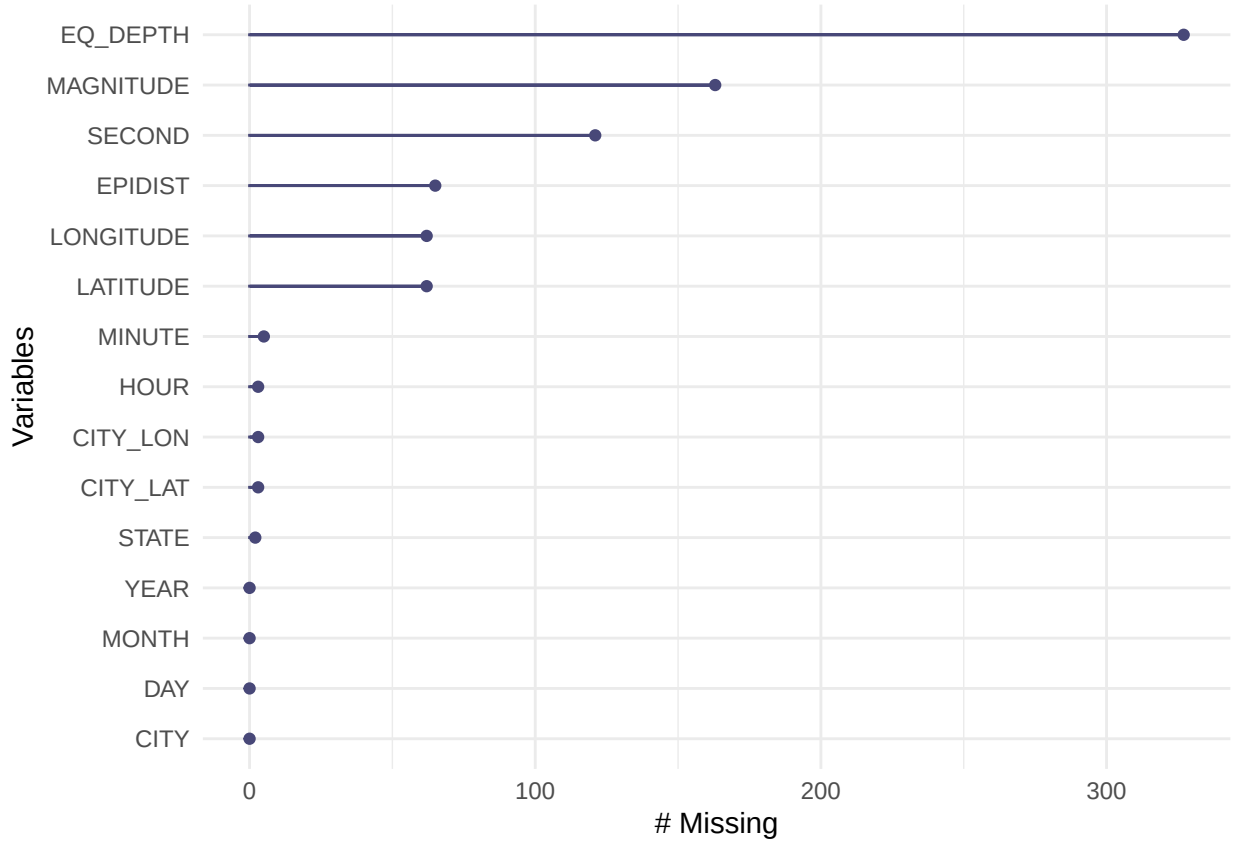
```
# Ottieni il numero di missing per partecipante (ne%)
miss_case_summary (df_1)
```

```
## # A tibble: 500 x 3
##     case n_miss pct_miss
##    <int>  <int>    <dbl>
##  1   121      8     53.3
##  2   276      8     53.3
##  3   424      8     53.3
##  4   317      7     46.7
##  5   375      7     46.7
##  6   464      7     46.7
##  7     1      6     40
##  8    11      6     40
##  9    58      6     40
## 10    65      6     40
## # ... with 490 more rows
```

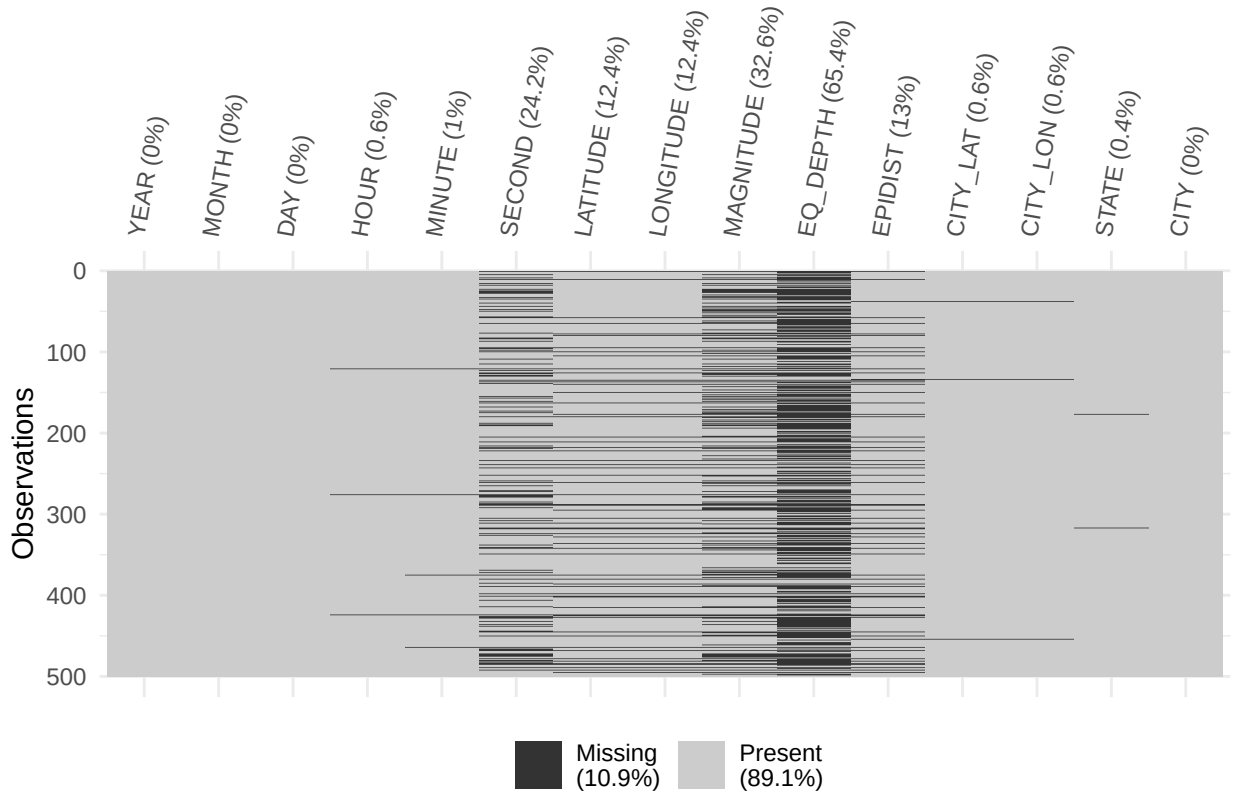
```
miss_case_table (df_1)
```

```
## # A tibble: 8 x 3
##   n_miss_in_case n_cases pct_cases
## *          <int>   <int>     <dbl>
## 1              0     161      32.2
## 2              1     166      33.2
## 3              2      50      10
## 4              3      61      12.2
## 5              5      14       2.8
## 6              6      42       8.4
## 7              7       3       0.6
## 8              8       3       0.6
```

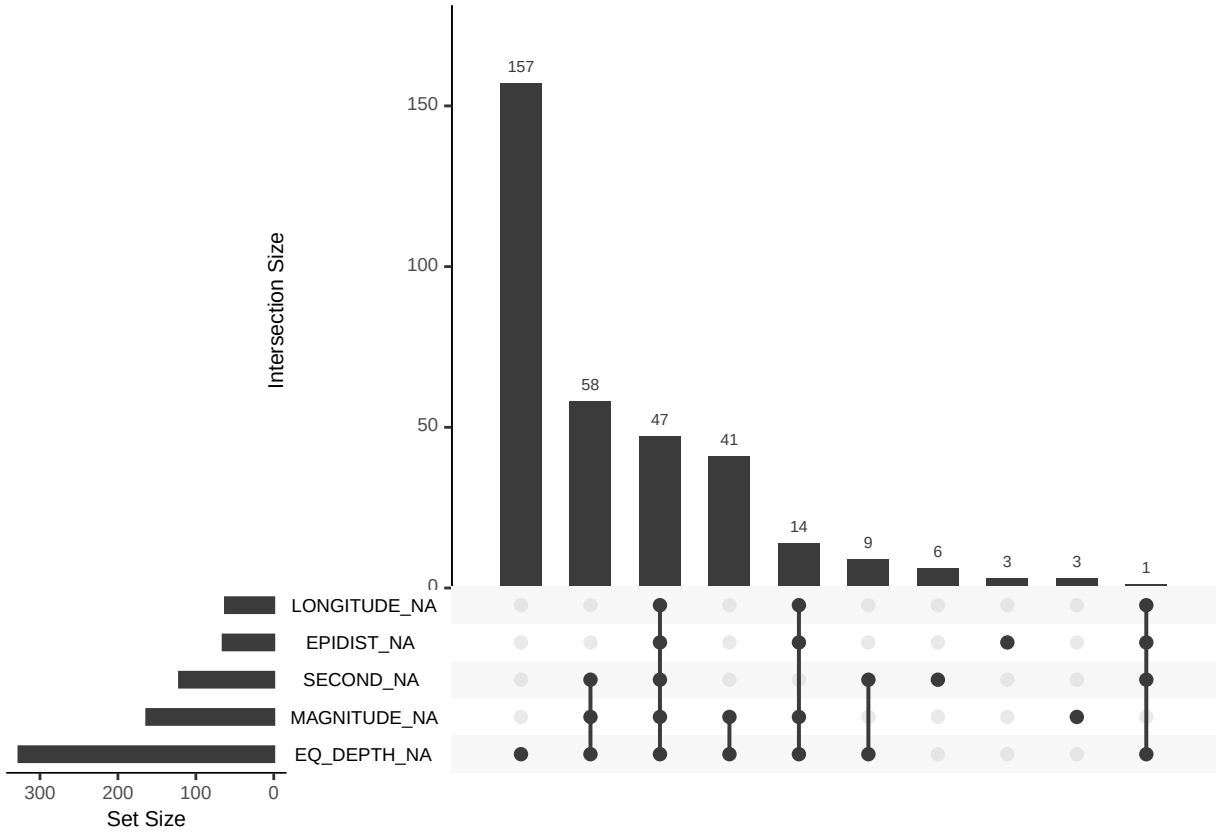
```
library(ggplot2)
# Quali variabili contengono le variabili piÃ¹ mancanti?
gg_miss_var (df_1)
```




```
# Dove si trovano gli oggetti mancanti?
vis_miss (df_1) + theme (axis.text.x = element_text (angle = 80))
```

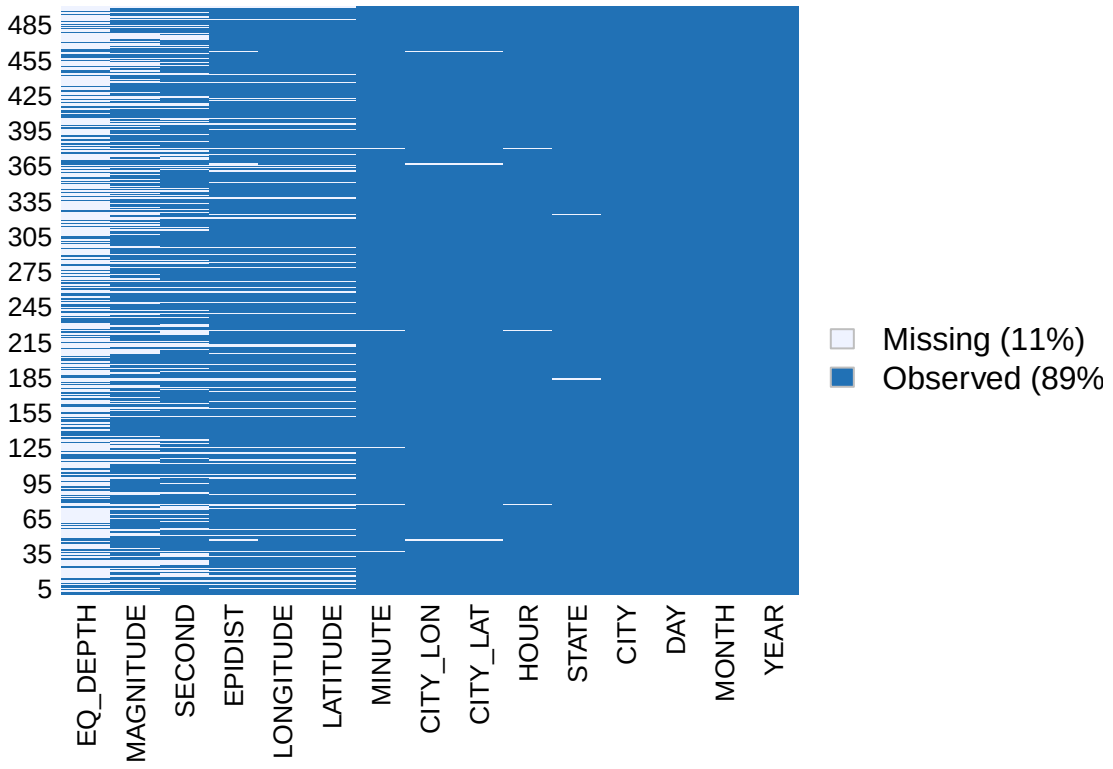



```
# Quali combinazioni di variabili mancano insieme?
gg_miss_upset (df_1)
```



Metodo Alternativo

```
# METODO ALTERNATIVO PER MOSTRARE I MISSING VALUE
library(Amelia)

missmap(df_1, main = "Earthwake Missing Values")
```

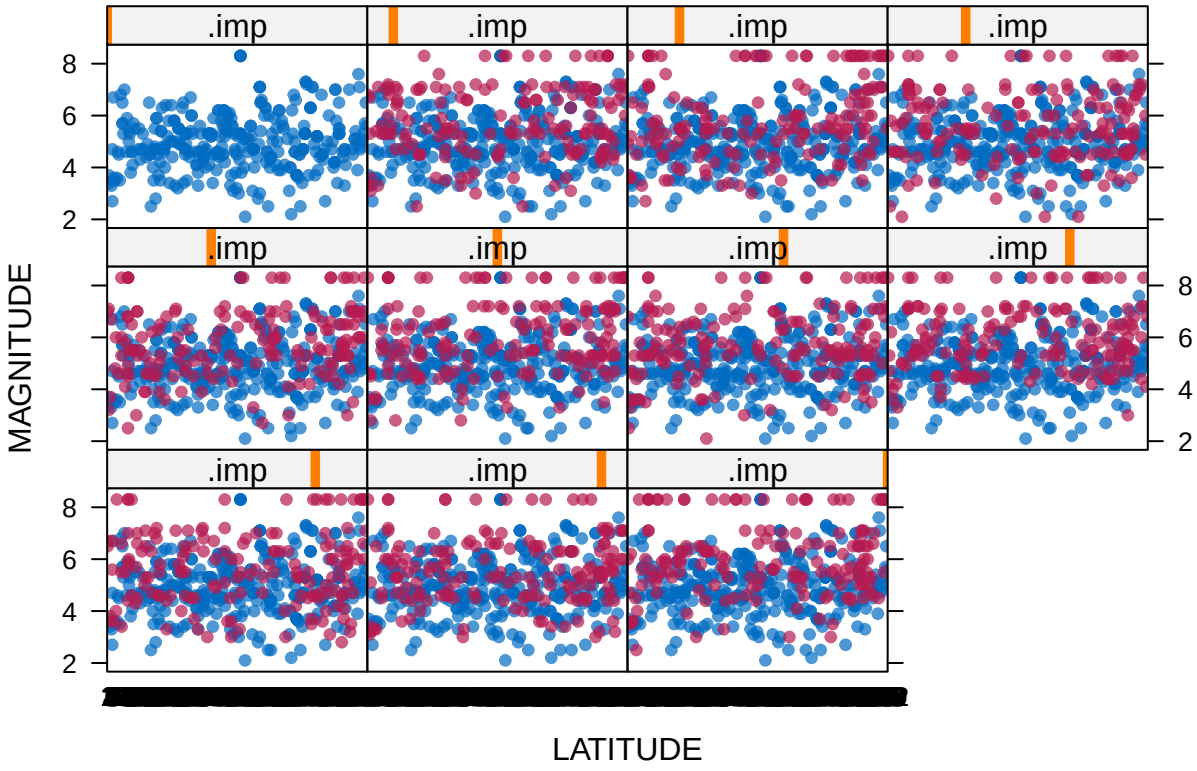


Imputazione Multivariata

```
# with
lm_multimp <- with (df_1_multimp, lm (MAGNITUDE ~ LATITUDE + LONGITUDE))
# pool
lm_pooled <- pool (lm_multimp)

summary(lm_pooled, conf.int = TRUE, conf.level = 0.95)
```

```
##          term    estimate   std.error statistic        df      p.value
## 1 (Intercept) 4.266505141 0.478201341 8.9219849 100.68923 2.176037e-14
## 2    LATITUDE 0.029039944 0.010595334 2.7408240  37.83012 9.303058e-03
## 3   LONGITUDE 0.001593587 0.002459608 0.6479027  55.95922 5.196975e-01
##          2.5 %      97.5 %
## 1  3.317846820 5.215163462
## 2  0.007587649 0.050492240
## 3 -0.003333680 0.006520854
```

```
stripplot (df_1_multimp,
           MAGNITUDE ~ LATITUDE | .imp,
           pch = 20, cex = 1)
```




```
df <-mice :: complete (df_1_multimp, 1)

library(tidyverse)
# Dividi i dati in training e set di test
set.seed (123)
0.8 * nrow (df)
```

```
## [1] 400
```

```
training.indices <- sample (1: nrow (df), 345)
train.data <- df [training.indices,]
test.data <- df [-training.indices,]
# Costruisci il modello del
model <- lm (MAGNITUDE ~ LATITUDE + LONGITUDE, data = train.data)
# Riepiloga il modello
summary(model)
```

```
##
## Call:
## lm(formula = MAGNITUDE ~ LATITUDE + LONGITUDE, data = train.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1684 -0.8076 -0.0736  0.8924  3.2769
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 4.106263   0.505111   8.129    8e-15 ***
## LATITUDE    0.035243   0.009866   3.572 0.000405 ***
## LONGITUDE   0.002472   0.002298   1.076 0.282891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.254 on 342 degrees of freedom
## Multiple R-squared:  0.04409,    Adjusted R-squared:  0.0385
## F-statistic: 7.887 on 2 and 342 DF,  p-value: 0.0004481
```

```
# Fai previsioni
predizioni <- model%>% predict(test.data)
# calcola l'errore di previsione, RMSE
sqrt (mean (model $ residuals ^ 2))
```

```
## [1] 1.24899
```

Principal Component Analysis

```
df_1[] <- lapply(df_1, function(x) {
  if(is.factor(x)) as.numeric(as.character(x)) else x
})
sapply(df_1, class)
```

```
##        YEAR       MONTH         DAY        HOUR      MINUTE      SECOND
##   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"   "numeric"
##    LATITUDE   LONGITUDE   MAGNITUDE    EQ_DEPTH     EPIDIST    CITY_LAT
##   "numeric"   "numeric"   "numeric"   "numeric" "character"   "numeric"
##    CITY_LON       STATE        CITY
##   "numeric" "character" "character"
```

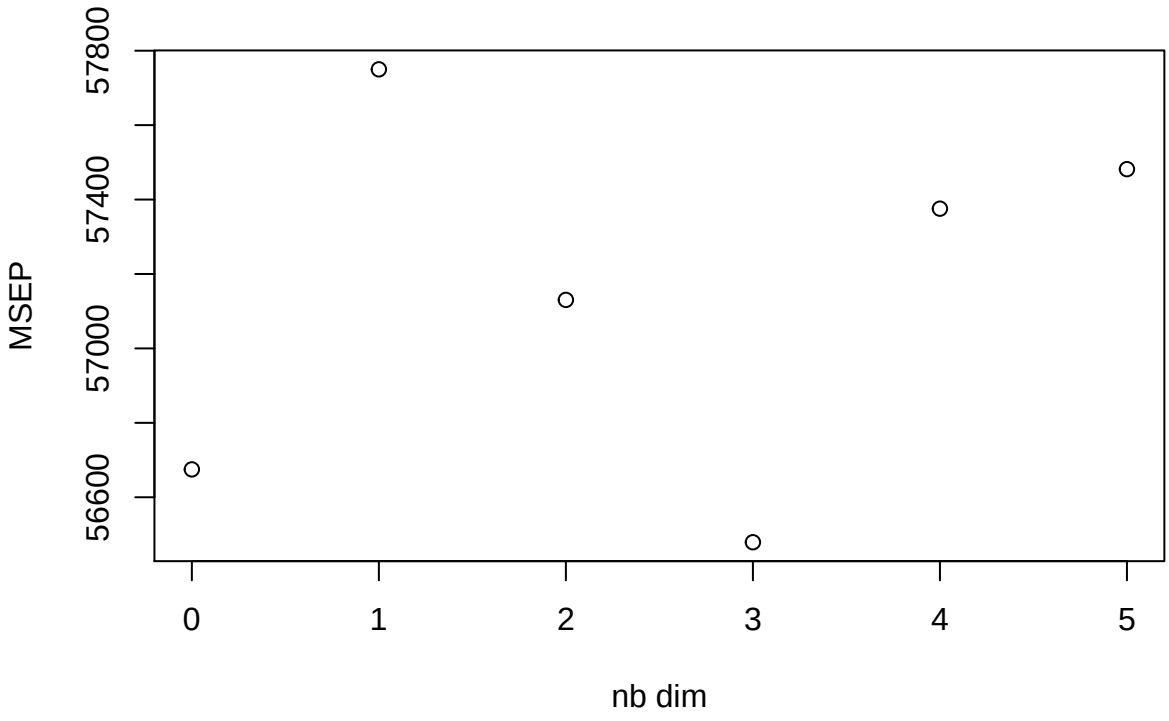
```
df_3= df_1[,1:10]
#PCA with missing values
library(missMDA)
```

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

```
# estim_ncpPCA = Estimate the number of dimensions for the Principal Component
#Analysis by cross-validation
# imputePCA= Impute dataset with PCA

nb <- estim_ncpPCA(df_3,method.cv = "Kfold", verbose = FALSE)
# estimate the number of components from incomplete data
#(available methods include GCV to approximate CV)
nb$ncp
```

```
## [1] 3
```

```
plot(0:5, nb$criterion, xlab = "nb dim", ylab = "MSEP")
```



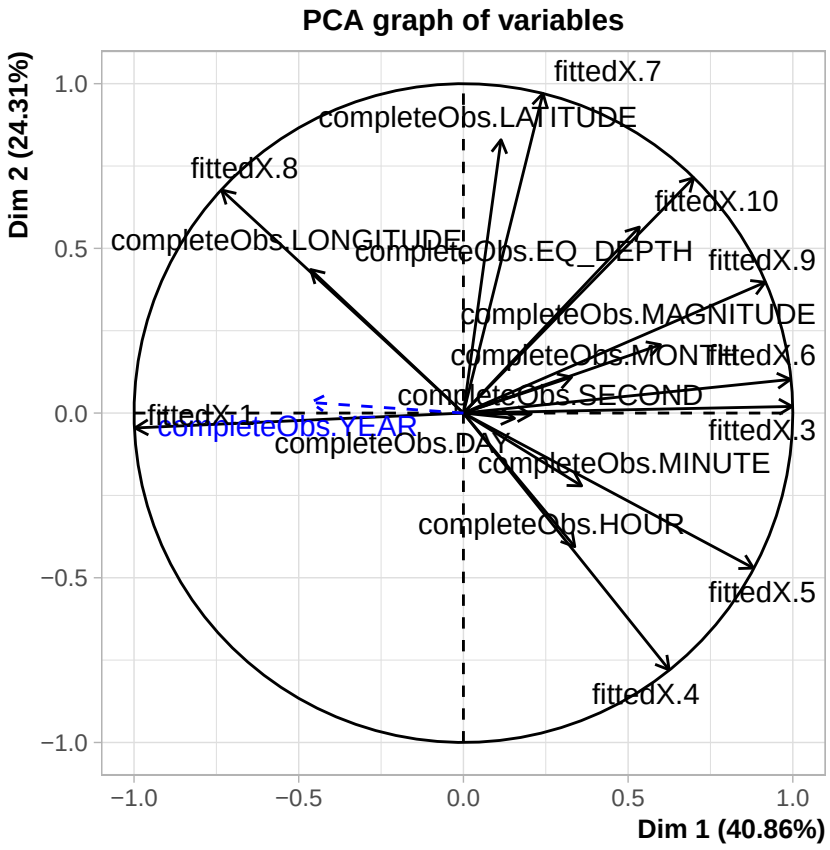

```
res.comp <- imputePCA(df_3, ncp = nb$ncp)
# iterativePCA algorithm
res.comp$completeObs[1:3,]
```

```
##      YEAR MONTH DAY HOUR MINUTE   SECOND LATITUDE LONGITUDE MAGNITUDE EQ_DEPTH
## [1,] 1931     8  16    3     40 30.00381 40.41303 -111.0093  5.239177 19.60601
## [2,] 1959    12  29    2     32 53.00000 36.90000 -121.4800  4.700000 20.41595
## [3,] 1954     1  27   14     19 48.00000 35.15000 -118.6300  5.000000 13.88632
```

```
# the imputed data set
imp <- cbind.data.frame(res.comp$completeObs,df_3)
df_4= imputePCA(df_3)
library(FactoMineR)
```

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

```
res.pca <- PCA(df_4, quanti.sup = 1, quali.sup = 12, ncp = nb$ncp, graph=FALSE)
# plot(res.pca, hab=12, lab="quali")
plot(res.pca, choix="var")
```




```
head(res.pca$ind$coord) #scores (principal components)
```

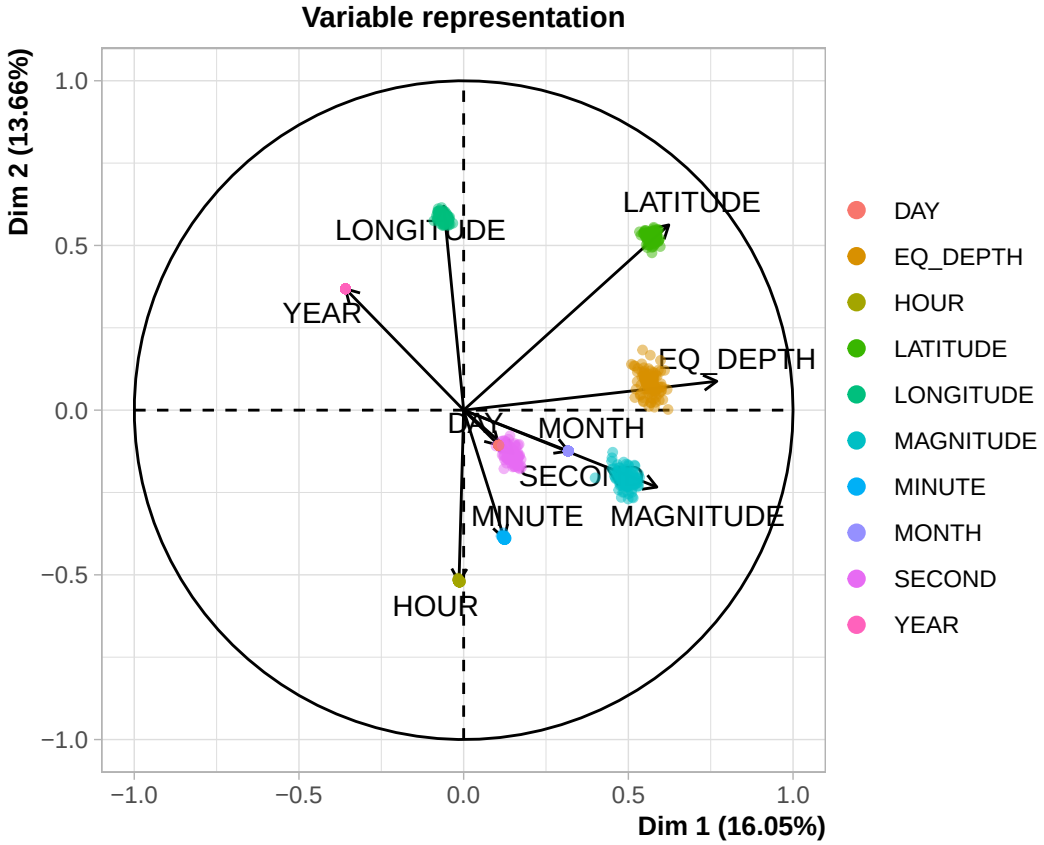
```
##        Dim.1      Dim.2     Dim.3
## 1  1.0275142  0.8649378 0.8312767
## 2  0.9559802  0.2111578 2.6559898
## 3 -0.2225353 -1.4162850 0.4500165
## 4  2.1285182 -4.4686974 0.4237243
## 5 -0.7543435 -0.1958332 0.3719395
## 6  1.2459785 -0.6517748 0.7359213
```

```
# Multiple imputation
library(Amelia)
# amelia= Multiple Imputation of Incomplete Multivariate Data
res.amelia <- amelia(as.data.frame(df_3), m = 5)
```

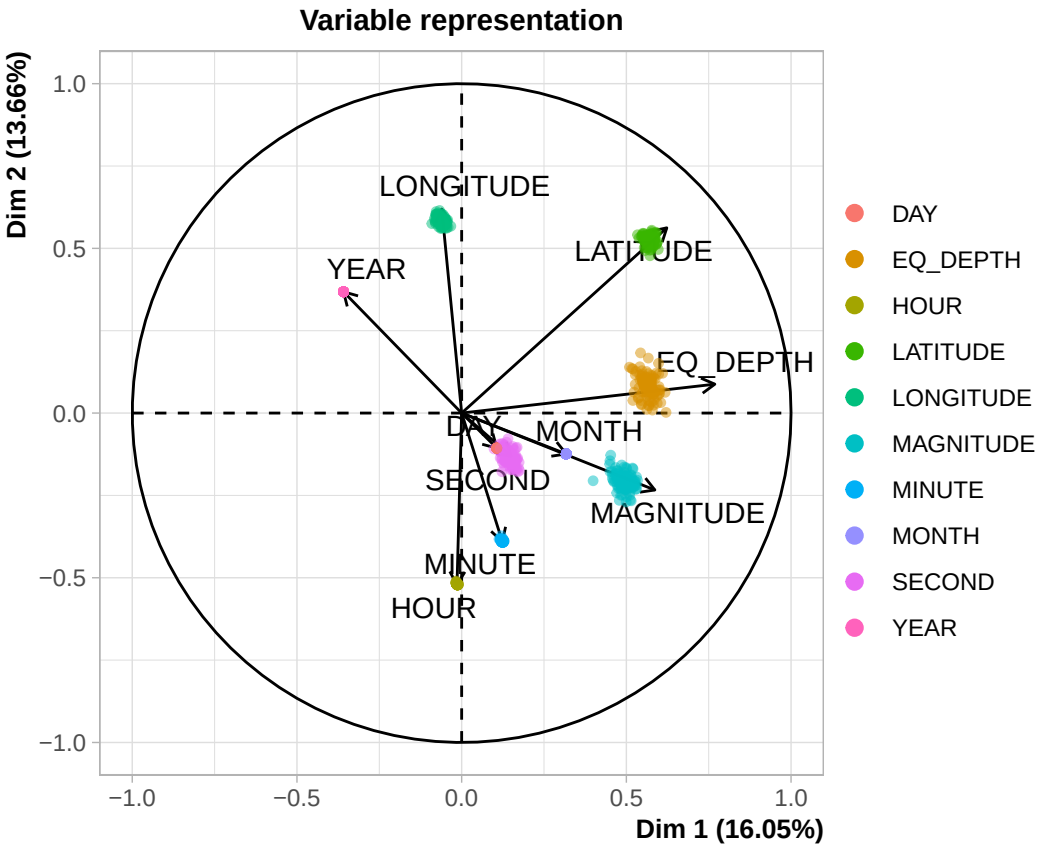
```
## -- Imputation 1 --
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
##
## -- Imputation 2 --
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  41 42 43
##
## -- Imputation 3 --
```

```
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  41 42 43 44 45 46
##
## -- Imputation 4 --
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
##  61 62 63 64 65
##
## -- Imputation 5 --
##
##   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
##  41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
```

```r
# the variability of the parameters is obtained
# MIPCA= Multiple Imputation with PCA

res.MIPCA <- MIPCA(df_3, ncp = 2, nboot = 100) # MI with PCA using 2 dimensions
#Inspect the imputed values

plot(res.MIPCA,choice= "var")
```




```
## $PlotVar
```




```
# Visualize the pattern
library(VIM)
```

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

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

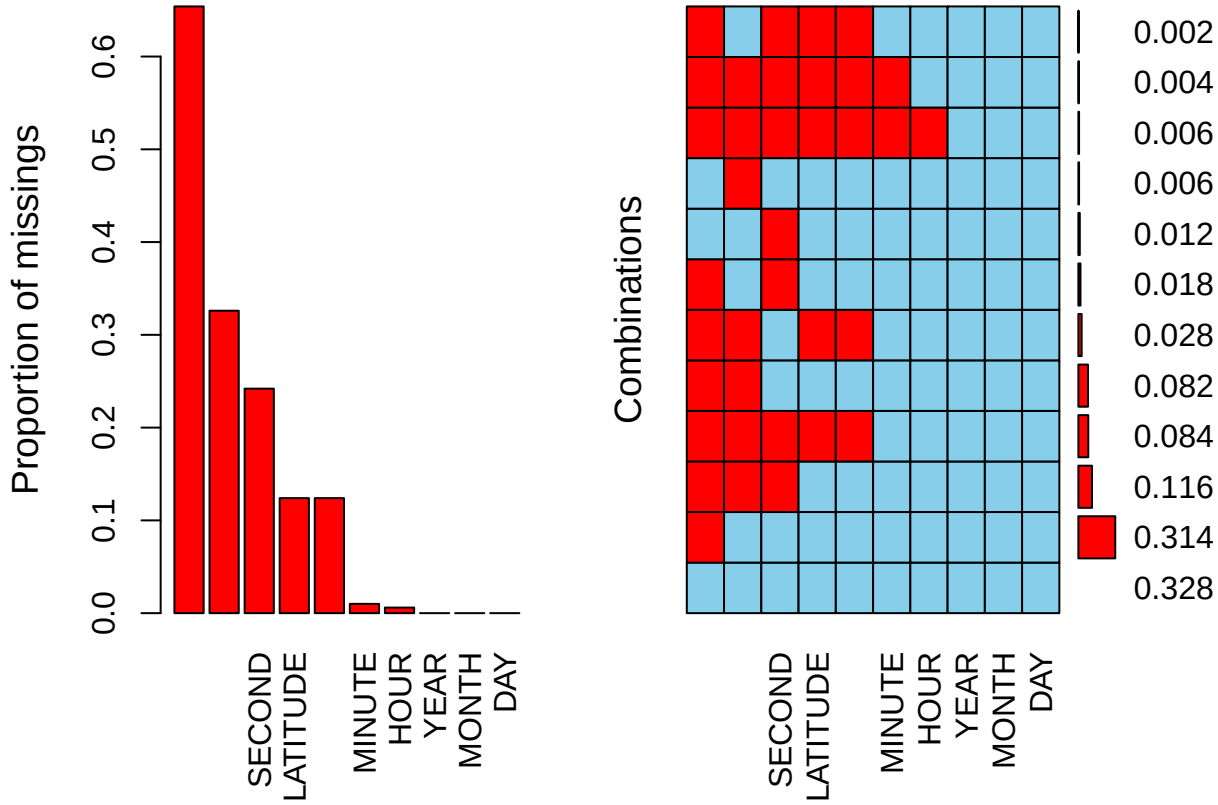
```
## VIM is ready to use.
```

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

```
##
## Attaching package: 'VIM'
```

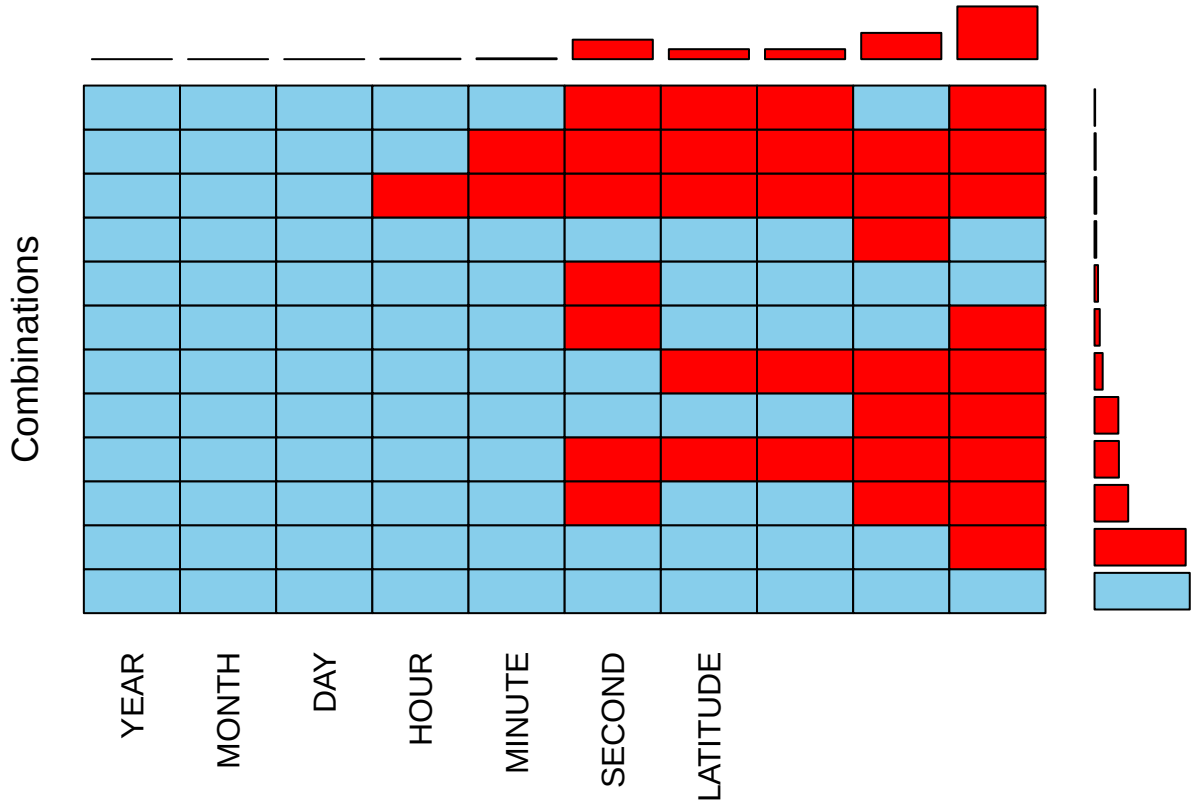
```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
aggr(df_3,only.miss=TRUE,numbers=TRUE,sortVar=TRUE)
```



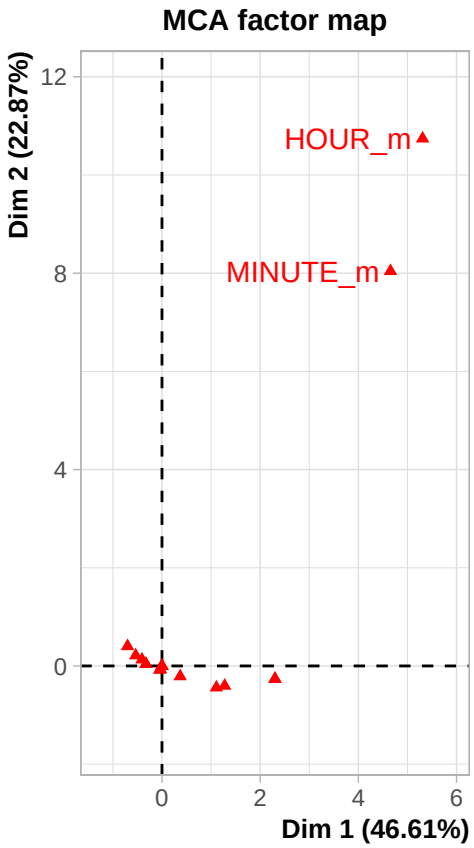

```
##
##  Variables sorted by number of missings:
##   Variable Count
##   EQ_DEPTH 0.654
##  MAGNITUDE 0.326
##     SECOND 0.242
##   LATITUDE 0.124
##  LONGITUDE 0.124
##     MINUTE 0.010
##       HOUR 0.006
##       YEAR 0.000
##      MONTH 0.000
##        DAY 0.000
```

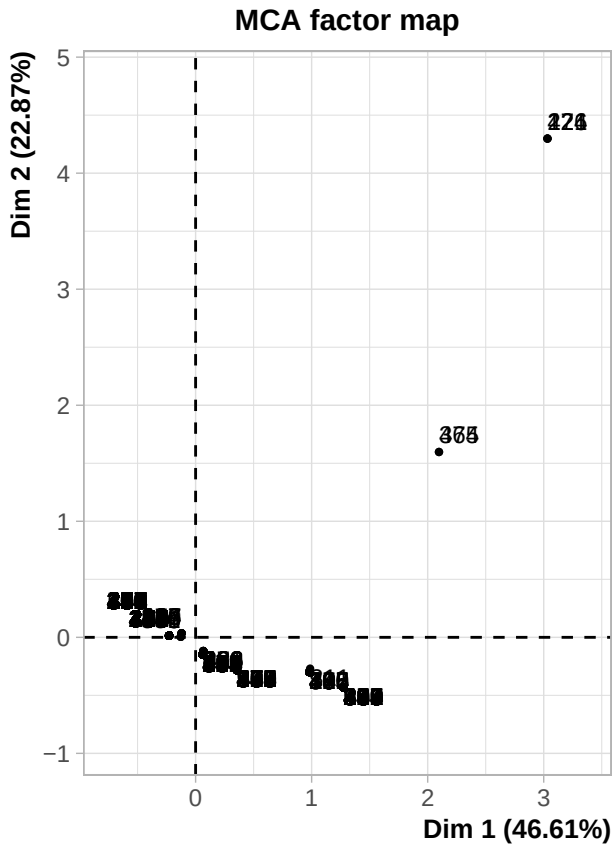
```
res <- summary(aggr(df_3,prop=TRUE,combined=TRUE))$combinations
```

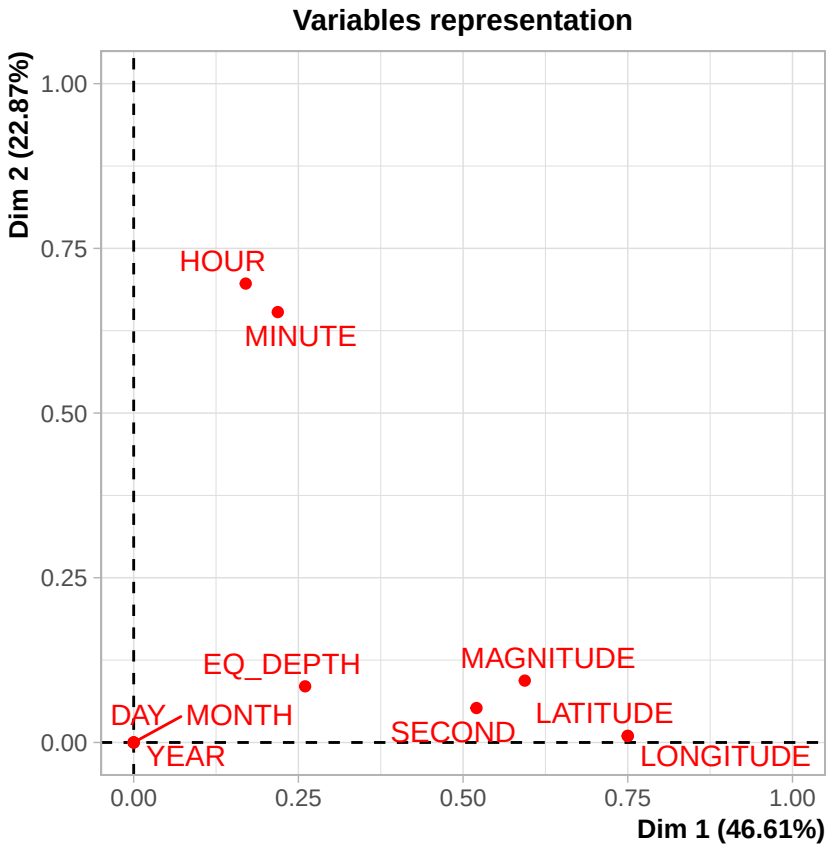



```
mis.ind <- matrix("o",nrow=nrow(df_3),ncol=ncol(df_3))
mis.ind[is.na(df_3)] <- "m"
dimnames(mis.ind) <- dimnames(df_3)
library(FactoMineR)
resMCA <- MCA(mis.ind)
```

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

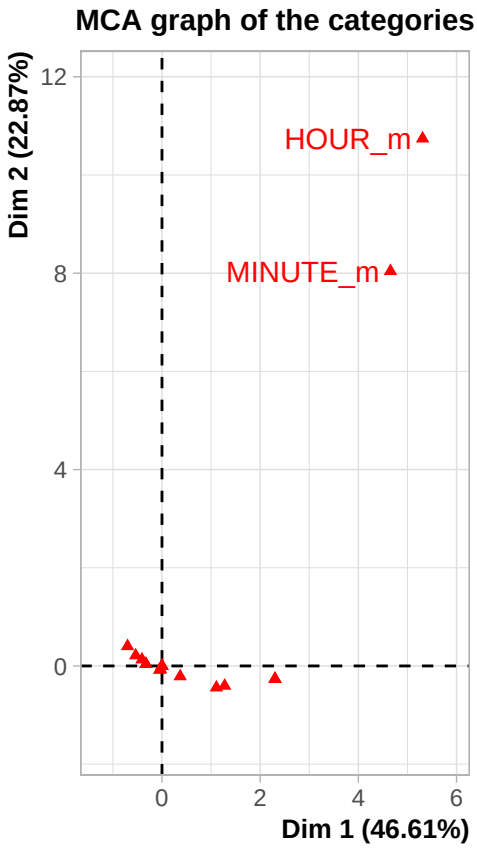

MCA factor map
Dim 2 (22.87%)
12
8
4
0
HOUR_m
MINUTE_m
0
2
4
6
Dim 1 (46.61%)


MCA factor map
Dim 2 (22.87%)
Dim 1 (46.61%)
5
4
3
2
1
0
−1
0
1
2
3




```
plot(resMCA,invis="ind",title="MCA graph of the categories")
```

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



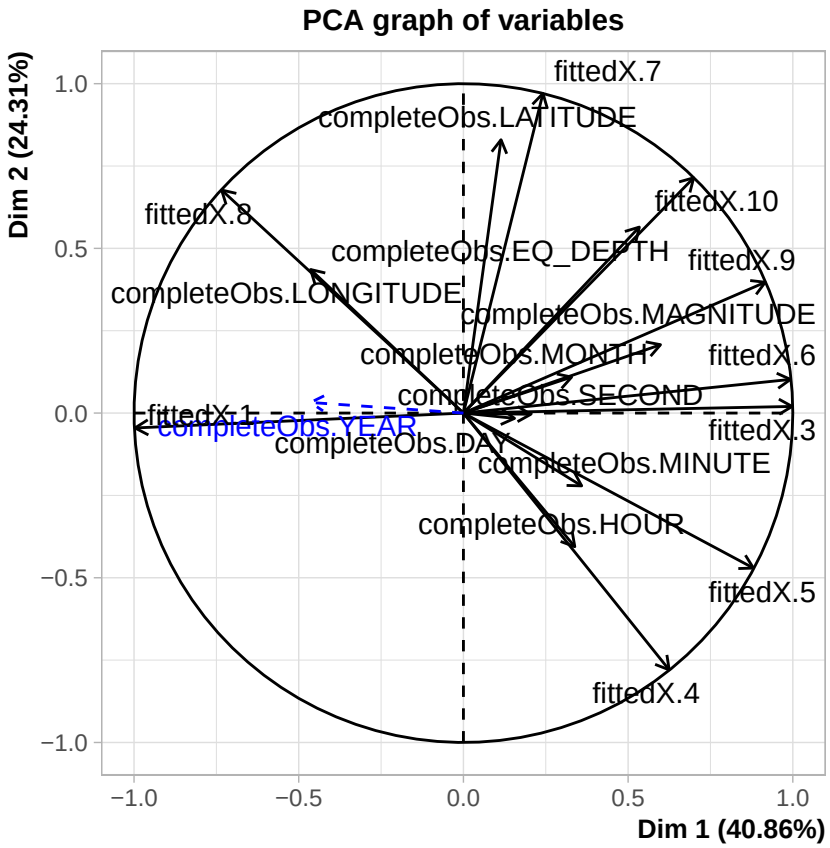

```
### Impute the incomplete data set
library(missMDA)
nb <- estim_ncpPCA(df_3,method.cv="Kfold",nbsim=100)
```

```
##   |                                                                      |
```

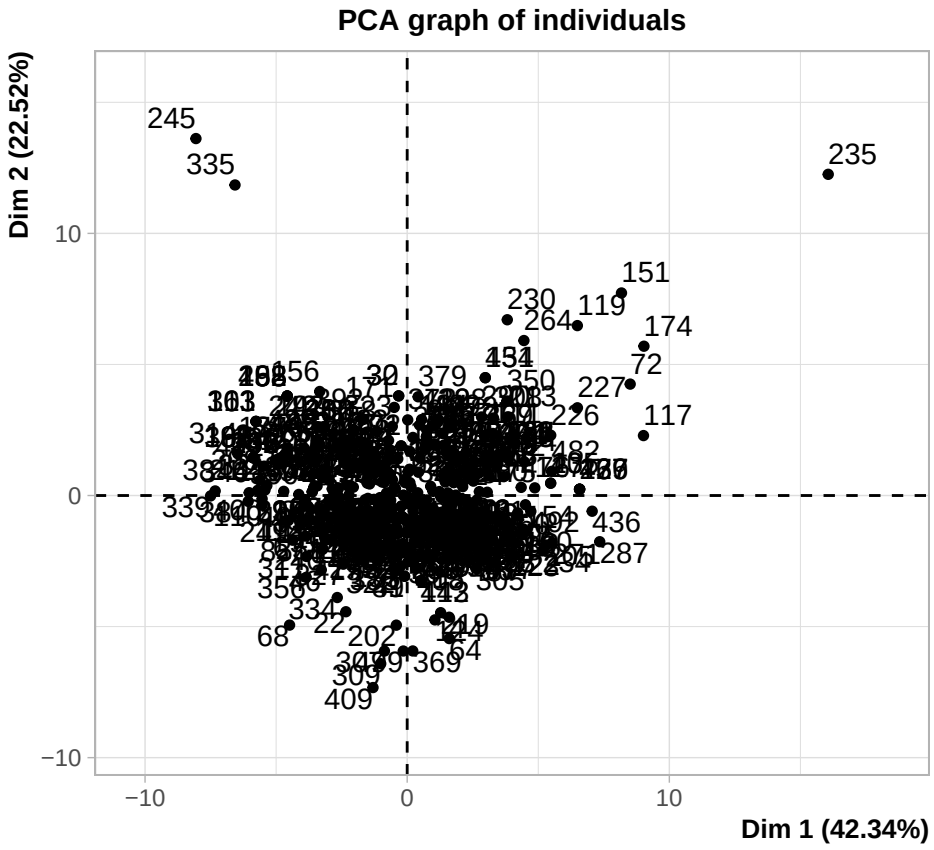
```
nb
```

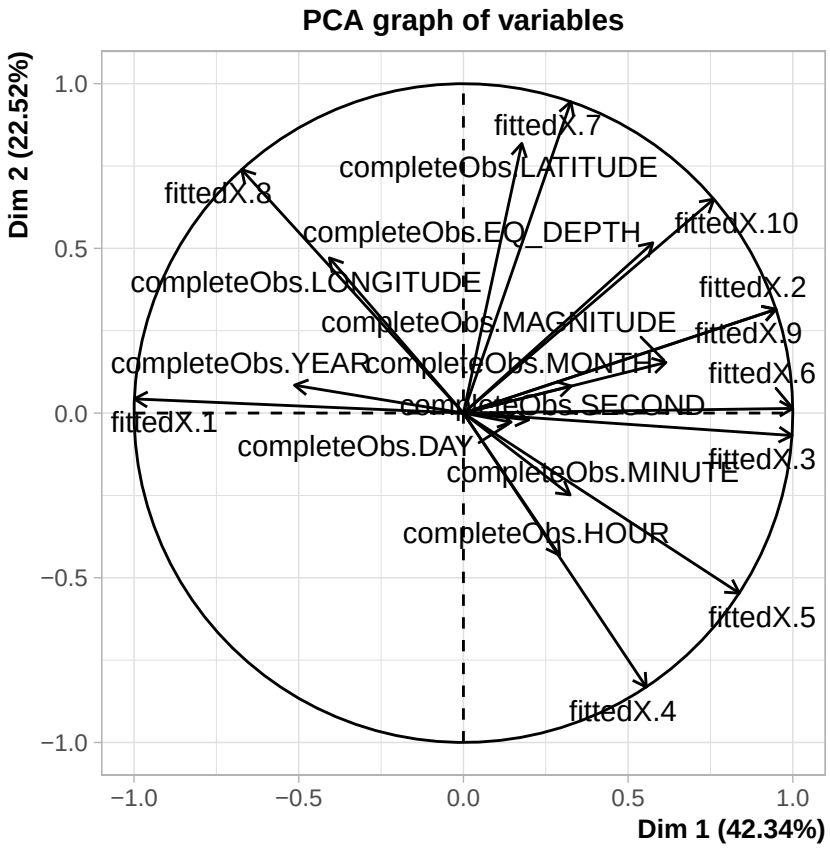
```
## $ncp
## [1] 0
##
## $criterion
##        0        1        2        3        4        5
## 53308.43 54314.21 54106.17 53467.99 54361.00 54640.96
```

```
res.comp <- imputePCA(df_3,ncp=2)

#Perform a PCA on the completed data set


plot(res.pca, choix="var")
```




```
# Compare with PCA on the data imputed by the mean
PCA(df_4)
```


PCA graph of individuals
Dim 2 (22.52%)
Dim 1 (42.34%)
245
335
235
151
230
119
264
174
72
227
117
226
436
287
68
369
409
10
0
−10
−10
0
10




```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 500 individuals, described by 20 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

```
#Categorical/mixed/multi-block data with missing values
```

Conclusione

Il dataset analizzato presentava molti valori NA che sono stati debitamente segnalati e visualizzati.