Manipulação de Dados - Parte IV

Paulo Henrique S. Guimarães

1 Data Wrangling

Data wrangling é uma etapa anterior ao *data mining* (ou *machine learning*). O termo data wrangling - também chamado de *data preparation* - significa preparação de dados. O conceito é relativamente recente e diz respeito ao ato de coletar, limpar, normalizar, combinar, estruturar e organizar os dados que serão analisados.

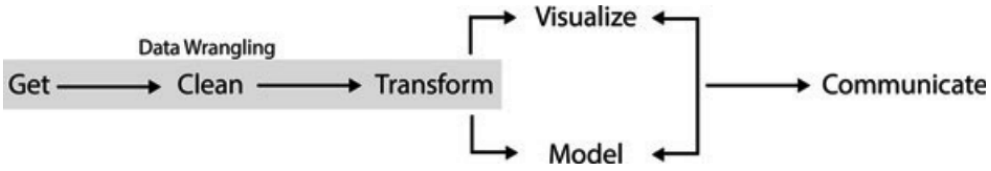

Figure 1: Data wrangling.

- Aplicação: exemplo dados de crédito

```
# Leitura da base de dados

credito = read.csv("creditdata.csv")
credito1<-read.csv("creditdata.csv")


# Resumo dos dados

summary(credito)
```

```
##     cliente          renda           idade          emprestimo
##  Min.   :   1.0   Min.   :20014   Min.   :-52.42   Min.   :    1.378
##  1st Qu.: 500.8   1st Qu.:32796   1st Qu.: 28.99   1st Qu.: 1939.709
##  Median :1000.5   Median :45789   Median : 41.32   Median : 3974.719
##  Mean   :1000.5   Mean   :45332   Mean   : 40.81   Mean   : 4444.370
##  3rd Qu.:1500.2   3rd Qu.:57791   3rd Qu.: 52.59   3rd Qu.: 6432.411
##  Max.   :2000.0   Max.   :69996   Max.   : 63.97   Max.   :13766.051
##                                   NA's   :3
##    pagamento
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1415
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

```r
# Apaga a coluna cliente

credito$cliente = NULL
credito1$cliente = NULL

# Valores inconsistentes

mean(credito$idade[credito$idade>0],na.rm = TRUE)
```

```
## [1] 40.9277
```

```r
credito$idade = ifelse(credito$idade < 0, 40.92,credito$idade)

mean(credito1$idade[credito1$idade>0],na.rm = TRUE)
```

```
## [1] 40.9277
```

```r
credito1$idade = ifelse(credito1$idade < 0, 40.92,credito1$idade)
```

1.0.1 Valores faltantes (Missing Data)

Existem várias maneiras de trabalharmos com valores omissos que podem ser maios ou menos adequadas dependendo da análise de dados a ser realizada. Uma possibilidade é excluir automaticamente as linhas de um *data frame* que tenha valores omissos, usando para tal a função **na.exclude (na.omit)**

Outra alternativa para resolver o problema de missing data é substituir por outros valores calculados de formas que possam fazer sentido, um processo chamado de imputação (substituir por exemplo, por zero, a média ou mediana do campo respectivo).

```r
credito$idade = ifelse(is.na(credito$idade), mean(credito$idade, na.rm = TRUE), credito$idade)
```

Outro métido mais "elaborado" é a imputação pelo método *missForest* que é uma implementação do algoritmo de *random forest*. É um método de imputação não paramétrico aplicável a vários tipos de variáveis.

- Exemplo:

```r
# install.packages("missForest")
library(missForest)
```

```
## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

## Loading required package: foreach

## Loading required package: itertools

## Loading required package: iterators
```

```r
data(iris)
iris.mis <- prodNA(iris, noNA = 0.1)
summary(iris.mis)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.10
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.30
##  Median :5.800   Median :3.000   Median :4.400   Median :1.30
##  Mean   :5.837   Mean   :3.032   Mean   :3.815   Mean   :1.22
##  3rd Qu.:6.400   3rd Qu.:3.275   3rd Qu.:5.100   3rd Qu.:1.80
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.50
##  NA's   :16      NA's   :12      NA's   :19      NA's   :11
##        Species
##  setosa    :44
##  versicolor:42
##  virginica :47
##  NA's      :17
##
##
##
```

```r
iris.imp <- missForest(iris.mis)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
```

```r
# iris.imp$ximp
```

- Voltando ao nosso conjunto de dados de crédito:

```r
library(missForest)

mean(credito1$idade) # por que aparece NA?
```

```
## [1] NA
```

```r
credito_idade<-missForest(credito1)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
```

```r
credito_idade<-credito_idade$ximp
mean(credito_idade$idade)
```

```
## [1] 40.93301
```

```
# Escalonamento

credito[, 1:3] = scale(credito[, 1:3])

# Encode para a classe

credito$pagamento = factor(credito$pagamento, levels = c(0,1))
```

1.0.2 Pacote janitor

Arrumar as planilhas manualmente pode ser uma tarefa bastante difíciel. O R não trabalha bem com espaços, acentos, caracteres pouco usuais e, em alguns casos, colunas ou linhas vazias. De forma a resolver estes problemas, podemos utilizar o pacote **janitor**.

Figure 2: Pacote janitor.

```
# install.packages("janitor")

library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(readxl)
dirty_data <- read_excel("dirty_data.xlsx",
              col_types = c("text", "text", "text",
                "text", "numeric", "numeric", "text",
                "numeric", "text", "text", "numeric"))
```

```
## New names:
## * Certification -> Certification...9
## * Certification -> Certification...10
## * Certification -> Certification...11
```

```
head(dirty_data)
```

```
## # A tibble: 6 x 11
##   `First Name` `Last Name` `Employee Statu~ Subject `Hire Date`
##   <chr>        <chr>       <chr>            <chr>         <dbl>
## 1 Jason        Bourne      Teacher          PE            39690
## 2 Jason        Bourne      Teacher          Drafti~       39690
## 3 Alicia       Keys        Teacher          Music         37118
## 4 Ada          Lovelace    Teacher          <NA>          27515
## 5 Desus        Nice        Administration   Dean          41431
## 6 Chien-Shiung Wu          Teacher          Physics       11037
## # ... with 6 more variables: `% Allocated` <dbl>, `Full time?` <chr>, `do
## #   not edit! --->` <dbl>, Certification...9 <chr>,
## #   Certification...10 <chr>, Certification...11 <dbl>
```

```
# nomes atuais das colunas
names(dirty_data)
```

```
##  [1] "First Name"         "Last Name"          "Employee Status"
##  [4] "Subject"            "Hire Date"          "% Allocated"
##  [7] "Full time?"         "do not edit! --->"  "Certification...9"
## [10] "Certification...10" "Certification...11"
```

```
# corrigindo os espaços e salvando em um novo banco (dirty_data2)

dirty_data2<-clean_names(dirty_data)
names(dirty_data2)
```

```
##  [1] "first_name"        "last_name"         "employee_status"
##  [4] "subject"           "hire_date"         "percent_allocated"
##  [7] "full_time"         "do_not_edit"       "certification_9"
## [10] "certification_10"  "certification_11"
```

No banco de dados do exemplo existem linhas e colunas vazias. Para remover isso utilizaremos o comando "remove_empty", especificando linhas ("rows") e colunas ("cols") no argumento:

```
dirty_data3<-remove_empty(dirty_data2, which = c("rows", "cols"))
head(dirty_data3)
```

```
## # A tibble: 6 x 9
##   first_name last_name employee_status subject hire_date percent_allocat~
##   <chr>      <chr>     <chr>           <chr>       <dbl>            <dbl>
## 1 Jason      Bourne    Teacher         PE          39690             0.75
## 2 Jason      Bourne    Teacher         Drafti~     39690             0.25
## 3 Alicia     Keys      Teacher         Music       37118             1
## 4 Ada        Lovelace  Teacher         <NA>        27515             1
## 5 Desus      Nice      Administration  Dean        41431             1
## 6 Chien-Shi~ Wu        Teacher         Physics     11037             0.5
## # ... with 3 more variables: full_time <chr>, certification_9 <chr>,
## #   certification_10 <chr>
```

2 Referência

1) Livro: R for Data Science - Hadley Wickham & Garrett Grolemund. Alta Books, 2019.

2) Livro: Data Wrangling with R - Bradley C. Boehmke. Springer, 2016.

3) https://www.curso-r.com/blog/2017-07-24-janitor/

4) https://medium.com/bio-data-blog/veja-como-arrumar-a-bagun%C3%A7a-na-planilha-de-dados-c09ce4279cf

5) https://github.com/sfirke/janitor