Dealing with Missing Data

No dia a dia do profissional da área de dados é comum lidar com bases contendo nulos. Mas você sabe o que fazer em cada situação?

Ignorar, excluir, imputar são algumas opções para resolver este problema.

Vamos checar alguns exemplos e como cada opção nos afeta.

```
df <- read_csv("~/Documentos/base_alunos.csv")
```

```
## Rows: 222 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): materia
## dbl (3): id_aluno, nota_bimestre, qtd_faltas
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(df)
```

```
## Rows: 222
## Columns: 4
## $ id_aluno      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ materia       <chr> NA, "Língua Portuguesa", "História", "Geografia", "Ciênc~
## $ nota_bimestre <dbl> 8, 7, 6, 10, 7, 9, 6, 8, 10, 8, 7, NA, 9, 6, 9, 7, 10, 7~
## $ qtd_faltas    <dbl> 2, 1, 6, NA, 3, NA, 2, 1, NA, 2, 1, 3, NA, 4, NA, 2, NA,~
```

Notas média dos alunos por matéria

Temos uma base com notas e faltas de alunos de acordo com as disciplinas, vamos olhar a média das notas e faltas.

```
df %>%
  group_by(materia) %>%
  summarize(media_nota  = mean(nota_bimestre),
            media_falta = mean(qtd_faltas))
```

```
## # A tibble: 6 x 3
##   materia           media_nota media_falta
##   <chr>                  <dbl>       <dbl>
## 1 Ciências               NA             NA
## 2 Geografia              NA             NA
## 3 História               NA             NA
## 4 Língua Portuguesa      NA             NA
## 5 Matemática              7.57          NA
## 6 <NA>                   NA             NA
```

O resultado saiu quase todo em branco, pois quando há nulos não tem como calcular.

Alguma informação combinado com nulo sempre dará nulo.

Ignorando nulos

Podemos então usar o parâmetro da função mean que ignora os nulos.

```
df %>%
  group_by(materia) %>%
  summarize(media_nota  = mean(nota_bimestre, na.rm = T),
            media_falta = mean(qtd_faltas, na.rm = T))
```

```
## # A tibble: 6 x 3
##   materia           media_nota media_falta
##   <chr>                  <dbl>       <dbl>
## 1 Ciências                7.6         1.90
## 2 Geografia               7.56        2.5
## 3 História                7.93        2.75
## 4 Língua Portuguesa       7.53        1.84
## 5 Matemática              7.57        1.63
## 6 <NA>                    8           2.25
```

Mas ainda assim não ficou legal, temos registros onde a disciplina está nula, nesse caso é melhor eliminar a linha toda quando não se sabe a disciplina.

Dropando linhas com nulos em uma determinada coluna

```
df %>%
  drop_na(materia) %>%
  group_by(materia) %>%
  summarize(media_nota  = mean(nota_bimestre, na.rm = T),
            media_falta = mean(qtd_faltas, na.rm = T))
```

```
## # A tibble: 5 x 3
##   materia           media_nota media_falta
##   <chr>                  <dbl>       <dbl>
## 1 Ciências                7.6         1.90
## 2 Geografia               7.56        2.5
## 3 História                7.93        2.75
## 4 Língua Portuguesa       7.53        1.84
## 5 Matemática              7.57        1.63
```

A função drop_na eliminou todos os registros onde a materia estava nula.

Caso uma coluna não seja informada, a função elimina qualquer linha que contenha nulos.

Então por que não usamos o drop_na sempre para eliminar todas as linhas que tenha qualquer nulo ao invés de repetir o parâmetro na.rm em todos os cálculos?

Dropando linhas com nulos em qualquer lugar do dataset

```
df %>%
  drop_na() %>%
  group_by(materia) %>%
  summarize(media_nota  = mean(nota_bimestre),
            media_falta = mean(qtd_faltas))
```

```
## # A tibble: 5 x 3
##   materia           media_nota media_falta
##   <chr>                  <dbl>       <dbl>
## 1 Ciências                6.83        1.74
## 2 Geografia               6.81        2.56
## 3 História                6.65        2.59
## 4 Língua Portuguesa       6.75        1.38
## 5 Matemática              7.31        1.63
```

Essa alternativa acaba distorcendo um pouco o cálculo, pois quando elimina uma linha inteira é eliminado linhas com informações.

Conferindo total de nulos por coluna

```
colSums(is.na(df))
```

```
##      id_aluno       materia nota_bimestre    qtd_faltas
##             0             5            21            72
```

Tem muitos registros com nulos nas faltas, vamos investigar um pouco mais esse campo

```
df$qtd_faltas %>% table(useNA = "always")
```

```
## .
##    1    2    3    4    5    6    7 <NA>
##   60   51   22    8    5    3    1   72
```

```
df %>%
  filter(is.na(qtd_faltas)) %>%
  select(nota_bimestre) %>% table(useNA = "always")
```

```
## nota_bimestre
##    7    8    9   10 <NA>
##    5   17   29   21    0
```

Não existem registros com 0 faltas, provavelmente o nulo representa 0 faltas.

E esses alunos sem faltas têm notas muito altas, e como estamos excluindo do cálculo está diminuindo muito a média das notas.

Então nesse exercício faz mais sentido usar o drop_na apenas na matéria, e usar o na.rm para remover pontualmente os nulos de cada cálculo isolado.

E quando entra a imputação?

Vamos pensar que no caso da nota quando está nula é porque o aluno não compareceu no dia da prova, e quando está zero é porque o aluno compareceu, mas errou todas as questões zerando a prova.

Ainda assim o aluno que não compareceu levou um zero, mas como está nulo e estamos desconsiderando o resultado final fica “melhor” com menos notas zero do que de fato tem.

Então nesse caso temos que imputar os nulos.

```
df <- df %>%
    mutate(nota_bimestre = replace_na(nota_bimestre, 0))

colSums(is.na(df))
```

```
##      id_aluno       materia nota_bimestre    qtd_faltas
##             0             5             0            72
```

Por fim chegamos no resultado abaixo

```
df %>%
  drop_na(materia) %>%
  group_by(materia) %>%
  summarize(media_nota  = mean(nota_bimestre, na.rm = T),
            media_falta = mean(qtd_faltas, na.rm = T))
```

```
## # A tibble: 5 x 3
##   materia           media_nota media_falta
##   <chr>                  <dbl>       <dbl>
## 1 Ciências                6.49        1.90
## 2 Geografia               7.05        2.5
## 3 História                7.39        2.75
## 4 Língua Portuguesa       6.16        1.84
## 5 Matemática              7.57        1.63
```