INFORMAÇÕES SUPLEMENTARES

PROCEDIMENTO PARA REVISÃO MANUAL

Devido ao volume de registros pareados, próximo a 100 mil, foi necessário adotar algumas estratégias para tornar o processo mais ágil, uma vez que apenas o primeiro autor foi responsável pela revisão manual.

Foi construída uma interface gráfica (em linguagem *visual basic for applications* do Microsoft Excel® - VBA) para auxiliar na revisão manual dos pares. Os registros foram então agrupados de acordo com determinadas características e então revisados manualmente.

Para agrupar os registros buscaram-se características que pudessem ser mensuradas objetivamente e que eram frequentes. Por exemplo, 13% dos pareamentos apresentavam ao menos um dos registros sem data de nascimento, 58% apresentavam o último nome representado por um rol de apenas 8 nomes (ex.: "Silva", "Santos" e etc), dentre outras características.

Por exemplo, registros com data de nascimento faltante, primeiro nome comum e primeiro e último nome da mãe diferentes compuseram um grupo. Dessa forma, o revisor concentrou-se em pontos específicos como sobrenome, no exemplo anterior, e assim tomar decisões de forma mais rápida.

O único grupo classificado automaticamente como "não par" foi o que apresentava a data de primeiros sintomas, no SINAN, superior a seis meses a data do óbito. O intervalo de seis meses foi adotado com margem de segurança para erros de digitação. Assim, foi possível classificar como "não par" 30% dos registros.

Terminado o procedimento, os pares duvidosos foram submetidos a um outro processo de agrupamento e revisão manual, os registros que permaneceram classificados como duvidosos foram então classificados como "não par".

CÁLCULO DE MEDIDAS DE SIMILARIDADE

Cálculo de medida de similaridade para nomes

Para o cálculo da medida de similaridade dos nomes foi utilizada a distância de edição de Levenshtein [1,2].

A distância de Levenshtein contabiliza o número de operações de deleções, inserções e substituições que são necessárias para converter um nome em outro. Para facilitar a comparação do quão diferente (ou similar) são dois nomes, a distância de Levenshtein é convertida em medida de similaridade conforme a equação a seguir [3,4].

$$MS\ (s1,s2) = 1 - (DL/MaxC) \qquad (1)$$

MS é a medida de similaridade e corresponde a um número entre 0 e 1, *s1* e *s2* correspondem aos dois nomes que estão sendo comparados, *DL* é a distância de Levenshtein e *MaxC* é o número de caracteres do nome com maior número de caracteres que estão sendo comparados (*s1* ou *s2*).

Cálculo alternativo de medida de similaridade para nomes

No entanto, ao comparar “Rafael P Silva” e “Rafael Pereira Silva”, a medida de similaridade é de 0,7. Para contornar o desempenho insatisfatório em nomes com abreviações, a forma de calcular a medida de similaridade foi modificada para:

$$MS(s1,s2) = 1 - \{[DL - (MaxC - MinC)]/MinC\} \quad (2)$$

MinC é o número de caracteres do nome com menor número de caracteres que estão sendo comparados (*s1* ou *s2*). Utilizando-se a modificação no método de cálculo de medida de similaridade proposto pelo estudo, a medida de similaridade entre os nomes “Rafael P Silva” e “Rafael Pereira Silva” é 1.

A modificação no método de cálculo de medida de similaridade proposta só foi utilizada quando o tamanho dos nomes (em relação ao número de caracteres) comparados apresentavam uma diferença maior que 3 caracteres. O critério foi adotado para evitar a perda de especificidade ao comparar nomes como “Maria Pereira Silva” e “Mariana Pereira Silva”.

Algoritmo alternativo para cálculo da medida de similaridade dos nomes, em linguagem vb.net

```
Public Function LvModf(ByVal s1 As String, ByVal s2 As String) As Single
    Dim n As Integer = s1.Length
    Dim m As Integer = s2.Length
    Dim d(n + 1, m + 1) As Integer
    Dim MaxC, MinC As Integer
    Dim res As Double

    ' A similaridade e defina como zero quando um dos nomes comparados não existe
    If n = 0 Then
       Return 0
    End If
    If m = 0 Then
       Return 0
    End If

    Dim i As Integer
    Dim j As Integer

    'Cálculo da distância de levenshtein
    For i = 0 To n
       d(i, 0) = i
    Next
    For j = 0 To m
       d(0, j) = j
    Next
    For i = 1 To n
       For j = 1 To m
          Dim cost As Integer
          If s2(j - 1) = s1(i - 1) Then
             cost = 0
          Else
             cost = 1
          End If
          d(i, j) = Math.Min(Math.Min(d(i - 1, j) + 1, d(i, j - 1) + 1), d(i - 1, j - 1) + cost)
       Next
```

```
    Next

    ' Define qual é o nome com maior número de caracteres
    If n > m Then
       MaxC = n
       MinC = m
    Else
       MaxC = m
       MinC = n
    End If

    'Define qual o cálculo da medida de similaridade que será empregado
    If (MaxC - MinC) > 3 Then
       Return 1 - ((d(n, m) - (MaxC - MinC)) / MinC)
    Else
       Return 1 - (d(n, m) / MaxC)
    End If

End Function
```

Medida de similaridade para datas de nascimento

Para a medida de similaridade da data de nascimento, foi utilizado um algoritmo que resulta em um número entre 0 e 1. Cada parte da data corresponde a uma parte do valor final, isto é, o ano de nascimento corresponde a 0,5, o mês a 0,17 e o dia a 0,33. Os valores foram atribuídos empiricamente e levam em consideração a probabilidade de serem iguais ao acaso.

A medida de similaridade para data de nascimento também levou em consideração alguns erros comuns de digitação. Por exemplo, a inversão entre mês e dia de nascimento, com ano de nascimento igual (por exemplo, 05/02/1986 e 02/05/1986), resulta em medida de similaridade de 0,95. O algoritmo completo é descrito abaixo.

Algoritmo para cálculo da medida de similaridade das datas de nascimento, em linguagem vb.net

```
'Avalia a concordância de data
'O formato de entrada das datas é "aaaammdd" (ex.: 19991231)
Public Function DataAval(ByVal D1 As String, ByVal D2 As String) As Single

    Dim Res As Single = 0
    Dim Check As Integer = 0
    Dim AnoIgual As Boolean = False

    'Verificar se a data é igual
    If D1 = D2 Then
       Return 1
       GoTo Sair1
    End If

    'Verifica se o ano de nascimento é igual
    If Mid$(D1, 1, 4) = Mid$(D2, 1, 4) Then AnoIgual = True

    'Verificar condições especiais
    If AnoIgual Then 'Se o ano é igual
       If Mid$(D1, 7, 2) = Mid$(D2, 5, 2) And Mid$(D2, 7, 2) = Mid$(D1, 5, 2) Then
          Return 0.95 'verifica se houve inversão de data
```

```
            GoTo Sair1
        End If

        For i = 5 To 8
            If Mid$(D1, i, 1) = Mid$(D2, i, 1) Then
                Check = Check + 1 'Conta dígitos iguais do dia e mês de nascimento
            End If
        Next

        If Check = 3 Then
            Return 0.88 ' Se o ano é igual e só tem um erro
            GoTo Sair1
        End If
    End If

    'Quando não há condições especiais pontua mês dia e ano de forma diferente
    '1,8 pontos com o dia
    '1,2 ponto com o mês
    '3 pontos com ano

    'Dia
    If Mid$(D1, 7, 2) = Mid$(D2, 7, 2) Then
        Res = Res + 1.8 ' O dia é igual
    Else
        If AnoIgual Then 'Em caso de ano igual
            If (Mid$(D1, 7, 2) - Mid$(D2, 7, 1)) = 1 Or (Mid$(D1, 7, 2) - Mid$(D2, 7, 1)) = -1
Then
                Res = Res + 0.6 ' Verifica diferença de mais ou menos 1
                GoTo SairA
            End If

            If Mid$(D1, 7, 1) = Mid$(D2, 8, 1) Then
                Res = Res + 0.8 ' Verifica inversão (ex.: 12 e 21)
                GoTo SairA
            End If
        End If

        If Mid$(D1, 7, 1) = Mid$(D2, 7, 1) Then ' 1 dígito correto
            Res = Res + 0.4
        End If

        If Mid$(D1, 8, 1) = Mid$(D2, 8, 1) Then ' 1 dígito correto
            Res = Res + 0.6
        End If
    End If

SairA:

    'Mês
    If Mid$(D1, 5, 2) = Mid$(D2, 5, 2) Then
        Res = Res + 1.2 ' O mês é igual
    Else
        If AnoIgual Then 'Em caso de ano igual
            If ((Mid$(D1, 5, 2) - Mid$(D2, 5, 2)) = 1 Or (Mid$(D1, 5, 2) - Mid$(D2, 5, 2)) =
-1) Then
                Res = Res + 0.5 ' Verifica diferença de mais ou menos 1
                GoTo SairB
            End If
            If Mid$(D1, 5, 1) - Mid$(D2, 6, 1) Then ' Verifica inversão
                Res = Res + 0.6 ' Verifica inversão (ex.: 10 e 01)
                GoTo SairB
            End If
        End If
```

```
            If Mid$(D1, 5, 1) = Mid$(D2, 5, 1) Then ' 1 dígito
               Res = Res + 0.3
            End If

            If Mid$(D1, 6, 1) = Mid$(D2, 6, 1) Then '1 dígito
               Res = Res + 0.4
            End If
         End If

SairB:

         'Ano
         'Detalha eventual erro na digitação de um dos dígitos do ano
         ' identifica diferença entre anos
         Dim AA As Integer = Mid$(D2, 1, 4) - Mid$(D1, 1, 4)
         Dim DifA As Boolean = False
         'Encontra apenas a diferença de um dígito (ex.: 1955 e 1956 ou 1965 e 1955| mas
não 1960 e 1966)
         If AA = 1 Or AA = -1 Or AA = 10 Or AA = -10 Then DifA = True

         'Processa ano
         If Mid$(D1, 1, 4) = Mid$(D2, 1, 4) Then
            If Res = 0 Then ' O ano é igual, mas o dia e mês não
               Res = 2
            ElseIf Res < 1 Then ' O ano é igual, mas o restante é bem diferente
               Res = 2.2
            Else
               Res = Res + 2.5
            End If
         ElseIf Mid$(D1, 3, 1) = Mid$(D2, 3, 1) Or Mid$(D1, 4, 1) = Mid$(D2, 4, 1) Or DifA
Then ' Apenas um dos dígitos do ano é igual
            If Res = 3 Then
               If DifA Then '(ex.: 1955 e 1956 ou 1965 e 1955)
                  Res = 4.9
               Else
                  Res = 4.5
               End If
            ElseIf DifA Then '(ex.: 1955 e 1956 ou 1965 e 1955)
               Res = Res + 1.8
            ElseIf Res > 2 Then
               Res = Res + 1.2
            Else
               Res = Res + 0.8
            End If
         End If

         'Interpreta
         If Res = 0 Then
            Return 0
         Else
            Return Res / 6
         End If

Sair1:
   End Function
```

Bibliografia

1. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl. 1966;10(8):707–10.

2. Bilenko M, Mooney RJ. Adaptive Duplicate Detection Using Learnable String Similarity Measures. 2003;39–48.

3. Camargo Jr KR, Coeli CM. Going open source: some lessons learned from the development of OpenRecLink. Cad Sa?de P?blica. 2015;31(2):257–63.

4. Christen P. Data matching : concepts and techniques for record linkage, entity resolution, and duplicate detection [Internet]. Springer; 2012.