Ministério do Planejamento, Orçamento e Gestão
**Instituto Brasileiro de Geografia e Estatística – IBGE**
Diretoria de Pesquisas
Coordenação de Métodos e Qualidade

Textos para discussão
Diretoria de Pesquisas
Número 41

# Pareamento Automático na Pesquisa de Avaliação da Cobertura da Coleta do Censo Demográfico de 2010

Djalma Galvão Carneiro Pessoa
Fábio Figueiredo Farias
Vinicius Layter Xavier

Rio de Janeiro
2012

**Instituto Brasileiro de Geografia e Estatística - IBGE**
Av. Franklin Roosevelt, 166 - Centro - 20021-120 - Rio de Janeiro, RJ - Brasil

**ISSN 1518-675X Textos para discussão. Diretoria de Pesquisas**

Divulga estudos e outros trabalhos técnicos desenvolvidos pelo IBGE ou em conjunto com outras instituições, bem como resultantes de consultorias técnicas e traduções consideradas relevantes para disseminação pelo Instituto. A série está subdividida por unidade organizacional e os textos são de responsabilidade de cada área específica.

ISBN 978-85-240-4235-5



**Impressão**

Gráfica Digital/Centro de Documentação e Disseminação de Informações - CDDI/IBGE, em 2012.

**Capa**

Gerência de Criação/CDDI

Pessoa, Djalma G. C.,
Pareamento automático na pesquisa de avaliação da cobertura da coleta do censo demográfico de 2010 / Djalma Galvão Carneiro Pessoa, Fábio Figueiredo Farias, Vinicius Layter Xavier. - Rio de Janeiro : IBGE, Coordenação de Métodos e Qualidade, 2012.
50 p. – (Textos para discussão. Diretoria de Pesquisas, ISSN 1518-675X ; n. 41)

Inclui bibliografia.
ISBN 978-85-240-4235-5

1. Brasil – Censo demográfico, 2010 – Avaliação. 2. Censo – Avaliação. 3. Amostragem (Estatística). 4. Crítica de dados. 5. População – Métodos estatísticos. I. Farias, Fábio Figueiredo. II. Xavier, Vinicius Layter. III. IBGE. Coordenação de Métodos e Qualidade. IV. Título. V. Série.

**Gerência de Biblioteca e Acervos Especiais** CDU 311.213.1(81)2010
RJ/IBGE/2012-02 DEM

Impresso no Brasil / Printed in Brazil

# Sumário

# Apresentação

Como parte das atividades do Censo Demográfico 2010, o IBGE realizou a Pesquisa de Avaliação da Cobertura da Coleta - PA, realizada por amostragem probabilística logo após o término da coleta do censo, em cada setor censitário selecionado para a amostra da PA. Como o próprio nome diz, o objetivo dessa pesquisa é avaliar a operação censitária no que se refere à cobertura da coleta em campo, por meio de estimativas de indicadores de omissão de domicílios e de pessoas.

Uma característica importante da Pesquisa de Avaliação é o confronto entre as unidades investigadas nos dois levantamentos, Censo e PA, para a identificação de coincidências e, assim, permitir a obtenção das estimativas de cobertura da coleta do Censo.

Em 2010, pela primeira vez no IBGE, foi utilizado um método automático de confronto e identificação de pares de registros de domicílios e de pessoas entre os dois levantamentos.

Este documento apresenta a descrição da etapa do sistema de pareamento automático usado. Constitui documentação detalhada e fundamental para a compreensão da execução das rotinas aplicadas, e é complementar ao relatório final da Pesquisa de Avaliação da Cobertura da Coleta do Censo 2010, por Silva, A.D. et all. (2011), que contém a descrição completa da metodologia da pesquisa, incluindo o planejamento, o plano amostral, as diversas etapas de apuração, os indicadores, os estimadores e alguns resultados.

**Sonia Albieri**
Coordenação de Métodos e Qualidade

# Introdução

Desde o Censo de 1970, o IBGE vem realizando pesquisa de avaliação da cobertura da coleta dos dados do censo. Pela primeira vez, em 2010, foi adotado pelo IBGE um processo automatizado de identificação de domicílios e pessoas correspondentes dentre os listados no Censo 2010 e na PA. Para isso, antes era utilizada a comparação visual de questionários e listagens. O sistema projetado pela Diretoria de Informática foi estruturado em três etapas:

- pareamento automático;
- pareamento assistido e
- reconciliação.

Neste documento, descrevemos a execução da primeira etapa do sistema de pareamento. Na Seção 2, apresentamos os conceitos básicos do pareamento probabilístico, adotado no sistema. Na Seção 3, descrevemos as funções da *library* `RecordLinkage` utilizadas no *script* de implementação do pareamento probabilístico. Nos anexos de 1 a 6 reproduzimos os programas do R com os passos utilizados na deduplicação e no pareamento de arquivos do Censo e da PA, de domicílios e de pessoas. As três etapas do sistema de pareamento estão descritas com detalhes no Relatório Final da Pesquisa de Avaliação (2011).

# 1- Conceitos Básicos sobre Pareamento Probabilístico

A ideia básica do pareamento probabilístico é calcular escores que dependam das probabilidades de concordância e discordância de variáveis escolhidas nos pares de registros. Para isto, são comparadas as probabilidades de haver concordância de variáveis para pares corretos e incorretos. As variáveis que discriminam pares corretos de incorretos devem concordar com maior frequência para os pares corretos do que para os incorretos. Para definir o escore a ser utilizado, precisamos introduzir a notação necessária.

## 1.1- Notação

Consideremos dois conjuntos $A$ e $B$ de entidades que podem ser, por exemplo, domicílios, pessoas, empresas e etc. com interseção não-vazia. Os elementos genéricos desses conjuntos serão designados por $a \in A, b \in B$ . Associados aos conjuntos $A$ e $B$ **,** observamos um conjunto de variáveis cujos valores são armazenados em arquivos $\alpha(A)$ e $\alpha(B)$ , respectivamente. Os registros correspondentes às entidades $a$ e $b$ serão denotados por $\alpha(a)$ e $\alpha(b)$ , respectivamente. Definiremos dois subconjuntos do conjunto $\alpha(A) \times \alpha(B)$ de pares de registros:

**pares corretos**:

$$M = \{(\alpha(a), \alpha(b)) | a = b\};$$

**pares incorretos**:

$$U = \{(\alpha(a), \alpha(b)) | a \neq b\}.$$

Definição: Uma função de comparação associa a cada par de registros um vetor de comparação:

$$C : (\alpha(a), \alpha(b)) \rightarrow \gamma,$$

onde o vetor de comparação tem dimensão igual ao número de variáveis usadas na comparação e, usualmente, tem elementos no conjunto $\{0,1\}$ .

Consideremos, por exemplo, três variáveis para a comparação de registros :

$\gamma_1$ - último nome do responsável;

$\gamma_2$ - primeiro nome do responsável;

$\gamma_3$ - nome da rua.

Um padrão simples de concordância é $\gamma = (1,0,1)$ , onde o componente 1 indica concordância e 0 discordância. Um padrão complexo de concordância seria:

$$\gamma = (0{,}66; 0; 0{,}80) .$$

### 1.2- Razão de Verossimilhança

Consideremos as seguintes probabilidades condicionais:

$P(\gamma|M)$ - Probabilidade condicional de dois registros terem um vetor de comparação $\gamma$ dado que são pares;

$P(\gamma|U)$ - Probabilidade condicional de dois registros terem um vetor de comparação $\gamma$ dado que não são pares.

Seja $P(M)$ a probabilidade de dois registros $(\alpha(a), \alpha(b))$ serem um par. Então pelo Teorema de Bayes temos:

$$P(M|\gamma) = \frac{P(\gamma|M)P(M)}{(P(\gamma))},$$

mas

$$P(\gamma) = P(\gamma|M)P(M) + P(\gamma|U)(1 - P(M)),$$

logo

$$P(M|\gamma) = \frac{1}{1 + \frac{P(\gamma|U)(1-P(M))}{P(\gamma|M)P(M)}}.$$

Pela equação cima, vemos que $P(M|\gamma)$ é uma função crescente da razão

$R(\gamma) = \frac{P(\gamma|M)}{P(\gamma|U)}$, denominada razão de verossimilhança, e que será o escore utilizado no pareamento de registros

### 1.3- Regra de Pareamento

Definiremos a seguinte regra de pareamento:

1. Ordene os vetores de comparação de registros segundo o valor da razão de verossimilhança $R(\gamma)$;
2. Escolha pontos de corte superior $W_1$ e inferior $W_2$ para $R(\gamma)$;
3. Declare pares corretos os elementos de $\alpha(A) \times \alpha(B)$ com valores de $R(\gamma)$ maiores que $W_1$ e incorretos os com valores menores que $W_2$ ponto de corte inferior.

Uma regra de pareamento $F$ associa a cada vetor de comparação $\gamma \in \Gamma$ uma de três ações:

$A_1$ - pareia;

$A_2$ - não decide e

$A_3$ - não pareia.

$$F : \Gamma \to \{A_1, A_2, A_3\}$$

A regra de decisão de Fellegi-Sunter $F(\gamma)$ é determinada pelos pontos de corte $W_1$ e $W_2$ :

$$F(\gamma)=\begin{cases} A_1 & se \quad R(\gamma)\geqslant W_1 \\ A_3 & se \quad R(\gamma)\leqslant W_2 \\ A_2 & c.c \end{cases}$$

Os valores de $W_1$ e $W_2$ são determinados a partir de valores escolhidos para para as probabilidades de erro:

- Erro de parear registros de entidades distintas;
- Erro de não parear registros de entidades iguais.

Estes erros correspondem aos erros do tipo 1 e 2 da Teoria de Testes de Hipóteses.

Consideremos $K$ variáveis de comparação e um vetor de comparação $\gamma=(\gamma_1,\ldots,\gamma_K)$ . Usualmente, supõe-se independência condicional:

$$P(\gamma|M)=P(\gamma_1|M)\ldots P(\gamma_K|M)$$

e

$$P(\gamma|U)=P(\gamma_1|U)\ldots P(\gamma_K|U).$$

Introduziremos a seguinte notação: $m_k=P(\gamma_k=1|M)$ e $u_k=P(\gamma_k=1|U)$ para $k=1,\ldots,K$ .

Sob a hipótese de independência condicional a razão de verossimilhança fica:

$$R(\gamma)=\frac{P(\gamma|M)}{P(\gamma|U)}=\frac{P(\gamma_1|M)\ldots P(\gamma_K|M)}{P(\gamma_1|U)\ldots P(\gamma_K|U)} \quad .$$

Usualmente, toma-se o logaritmo de $R(\gamma)$ na base 2, obtendo-se o escore:

$$\begin{aligned} \log_2(R(\gamma)) &= \log_2\left(\frac{P(\gamma_1|M)}{P(\gamma_1|U)}\right)+\ldots+\log_2\left(\frac{P(\gamma_K|M)}{P(\gamma_K|U)}\right) \\ &= \log_2\left(\frac{m_1}{u_1}\right)+\ldots+\log_2\left(\frac{m_K}{m_K}\right). \end{aligned}$$

Por exemplo, consideremos as seguintes probabilidades de concordância para as variáveis de domicílio:

**Tabela 1: Exemplo de Cálculo de Escores**

| Variável | $P(Conc\|M)$ | $P(Conc\|U)$ |
|---|---|---|
| PrimNomResp | 0,9 | 0,2 |
| UltNomResp | 0,9 | 0,2 |
| NomeRua | 0,8 | 0,3 |
| NumDom | 0,8 | 0,3 |
| Idade | 0,9 | 0,2 |
| Sexo | 0,8 | 0,3 |

considere o seguinte par de registros:

**Tabela 2: Exemplos de registros**

| PrimNomResp | UltNomresp | NomeRua | NumDom | Idade | Sexo |
|---|---|---|---|---|---|
| Carlos | Gomes | Aurora | 450 | 65 | M |
| Carlos | Gomes | Aurora | 450 | 70 | F |

Nesse exemplo temos $\gamma=(1,1,1,1,0,0)$ e

$$\begin{aligned}\log_2(R(\gamma)) &= \log_2((0,9/0,2))+\log_2((0,9/0,2))+\log_2((0,8/0,3))\\ &+ \log_2((0,8/0,3))+\log_2((1-0,9)/(1-0,2))\\ &+ \log_2((1-0,8)/(1-0,3))=2,36257\end{aligned}$$

Os valores de $m_k$ e $u_k$ não são conhecidos na prática e, portanto, não é possível calcular o escore R. Contudo, esses valores podem ser estimados usando-se o Algoritmo EM, que é um método geral de obtenção de Estimadores de Máxima Verossimilhança na presença de dados faltantes. A partir de um modelo paramétrico para os dados observados, são executadas iterações com dois passos, cada uma. No passo E, é feita a imputação dos dados faltantes e no passo M é obtido o estimador de máxima verossimilhança para os dados completados. O procedimento para quando é satisfeita uma condição de convergência fixada.

Consideremos o caso particular em que o vetor tem componentes 0 e 1, e vale a hipótese de independência condicional. Para cada par de registros, observa-se o vetor de comparação das variáveis. Os valores observados são resumidos na Tabela 3 de frequências das configurações de comparação $(\gamma)$ das variáveis. Para fixar ideias, consideremos a seguinte tabela de frequências para o caso de três variáveis:

**Tabela 3: Frequências das configurações de comparação de variáveis**

| Conf | V1 | V2 | V3 | Freq |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 24.620 |
| 2 | 0 | 0 | 1 | 5 |
| 3 | 0 | 1 | 0 | 961 |
| 4 | 0 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 | 56.454 |
| 6 | 1 | 0 | 1 | 147 |
| 7 | 1 | 1 | 0 | 14.880 |
| 8 | 1 | 1 | 1 | 251 |

A primeira linha da Tabela 3 indica que há discordância de todas as variáveis em 24.620 pares de registros. Desses 24.620 pares, quantos seriam pares corretos e quantos incorretos? O Algoritmo EM permite decompor cada frequência da Tabela 3 em duas parcelas, que estimam as frequências de pares corretos e incorretos. Para os dados acima, o Algoritmo EM fornece para cada configuração a decomposição:

$$FREQ_i = FM_i + FU_i .$$

**Tabela 4: Resultados da aplicação do Algoritmo EM**

| Conf | V1 | V2 | V3 | FREQ | FM | FU |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 24.620 | 231,58 | 24.388,41 |
| 2 | 0 | 0 | 1 | 5 | 4,99 | 0,00 |
| 3 | 0 | 1 | 0 | 961 | 268,38 | 692,61 |
| 4 | 0 | 1 | 1 | 0 | 0,00 | 0,00 |
| 5 | 1 | 0 | 0 | 56.454 | 11.885,71 | 0,00 |
| 6 | 1 | 0 | 1 | 147 | 146,99 | 0,00 |
| 7 | 1 | 1 | 0 | 14.880 | 13.627,54 | 1.252,45 |
| 8 | 1 | 1 | 1 | 251 | 251,00 | 0,00 |

A partir da Tabela 4, podemos estimar as probabilidades de interesse:

- $P(M)$ - probabilidade de dois registros serem um par correto;
- $P(U)$ - probabilidade de dois registros serem um par incorreto;
- $P(\gamma|M)$ - probabilidade de observar a configuração $\gamma$ dado que os dois registros formam um par correto;
- $P(\gamma|U)$ - probabilidade de observar a configuração $\gamma$ dado que os dois registros formam um par incorreto.

Para efeito de ilustração, utilizaremos os resultados da Tabela 4 para estimar as probabilidades:

1. $P(M)$ de um par ser correto:

$$\hat{P}(M) = \frac{\sum_{i=1}^{8} FM_i}{\sum_{i=1}^{8} FREQ_i} = 0{,}2714;$$

2. $m_2 = P(\gamma_2 = 1|M)$ de concordância de valores da variável V2, dado que os registros formam um par correto.

Para isso, dividimos a soma das frequências em que há concordância na variável $V_2$ na coluna $FM$ pela frequência total :

$$\hat{P}(\gamma_2 = 1|M) = \frac{(FM_3 + FM_4 + FM_7 + FM_8)}{\sum_{i=1}^{8} FM_i} = 0{,}5355.$$

A partir das estimativas acima é possível estimar a probabilidade de o par ser correto dado que foi observada a configuração . Usando o Teorema de Bayes:

$$P(M|\gamma) = \frac{P(M, \gamma)}{P(\gamma)} = \frac{P(\gamma|M)P(M)}{P(\gamma|m)P(M) + P(\gamma|U)P(U)}$$

Depois de estimar essa probabilidade a posteriori, podemos usar a seguinte regra de classificação de Bayes: classifique como correto o par de registros cujo valor de $P(M|\gamma)$ for alto. Essa regra coincide com a regra baseada em $R(\gamma)$ , definida anteriormente.

# 2 - Funções da *Library* `RecordLinkage` do R

Na implementação das ideias descritas, foi utilizada a library `RecordLinkage` do R. Para ilustrar o procedimento, usaremos como exemplo arquivos de domicílios de um mesmo setor do Censo e da Pesquisa de Avaliação. Queremos parear domicílios.

Suporemos que os dados então disponíveis nos *data frames* PA354390705000135 e CE354390705000135 relativos ao setor 354390705000135, para a PA e o Censo, respectivamente. Nos dois *data frames* foram observadas as variáveis:

- `ID_SETOR;`
- `ID_DOM;`
- `PRIM_NOME_RESP;`
- `MEIO_NOME_RESP;`
- `ULT_NOME_RESP;`
- `LOGR_TIPO;`
- `LOGR_TITULO_NOME;`
- `NUMERO_MODIFICADOR_COMPLEMENTO.`

Inicialmente, a library `RecordLinkage` do R é carregada:

```
> library(RecordLinkage)
```

A função `compare.linkage` da library `RecordLinkage` compara todos os pares de registros dos dois *data frames*, e cria um *data frame* cujo número de linhas é o produto dos números de linhas dos dois *data frames*.

Podemos escolher as variáveis e o critério usados na comparação. Fixando o argumento `exclude` da função `compare.linkage` podemos excluir as variáveis não usadas na comparação dos registros. Na comparação de valores de variáveis de texto, podemos usar vários critérios para medir a similaridade. Adotaremos:

- `LOGR_TIPO`: exato;
- `LOGR_TITULO_NOME`: Jaro-Winkler;
- `NUMERO_MODIFICADOR_COMPLEMENTO`: exato.

Na função, o critério de Jaro-Winkler é fixado por meio do argumento `strcmpfun`. As variáveis que usam o critério de Jaro-Winkler são especificadas pelo argumento `strcmp`, por um vetor contendo os números das colunas.

A função `compare.linkage` cria um *data frame* com as comparações dos pares de registros nos dois arquivos. Os argumentos da função `compare.linkage` são:

| | |
|---|---|
| `dataset` | Tabela de registros a ser deduplicada. *Data frame* ou matriz. |
| `dataset1` `dataset2` | Dois conjuntos de dados a serem pareados. |
| `blockfld` | Definição do campo de blocagem. Uma lista de inteiros ou de vetores de caracteres com índices de coluna ou `FALSE` para desabilitar blocagem . |
| `phonetic` | Determina o uso de um código fonético. Se `FALSE`, nenhum código fonético será usado; se `TRUE`, o código fonético será usado para todas as colunas; se for dado um vetor numérico ou de caracteres, o código fonético será usado para as colunas especificadas. |
| `phonfun` | Função do código fonético. |
| `strcmp` | Determina o uso de uma métrica de cadeia de caracteres. Usado da mesma maneira que `phonetic`. |
| `strcmpfun` | Função para comparar variáveis de texto definida pelo usuário. |
| `exclude` | Colunas a serem excluídas. Um vetor numérico ou de caracteres que especifica as colunas que devem ser excluídas na comparação. |
| `identity`, `identity1`, `identity2` | Vetores numéricos opcionais para identificar pares e não-pares. Em um processo de deduplicação, dois registros `dataset[i,]` e `dataset[j,]` são um par verdadeiro se e só se `identity[i,]==identity[j,]`. Em um processo de pareamento, dois registros `dataset1[i,]` e `dataset2[j,]` são um par verdadeiro se e só se `identity1[i,]==identity2[j,]`. |
| `n_match`, `n_non_match` | Número de pares corretos e incorretos desejados no resultado. |

A saída da função `compare.linkage` é uma lista com os seguintes componentes:

| | |
|---|---|
| `data` | Cópia dos registros convertida em *data frame.* |
| `pairs` | Padrões de comparação gerados. |
| `frequencies` | Para cada coluna incluída em `pairs`, a frequência média de valores (inverso do número de valores distintos). |

Na aplicação a seguir, são definidos os argumentos, pela ordem: `dataset1; dataset2` e pelo nome: `exclude; strcmp; strcmpfun.`

```
>    rpairsPACE.11<-compare.linkage(PA354390705000135,CE354390705000135    ,
exclude=1:5 , strcmp=7, strcmpfun = jarowinkler)
```

O objeto criado `rpairsPACE.11` é uma lista com os seguintes componentes:

```
"data1"    "data2"    "pairs"    "frequencies" "type".
```

O componente `pairs` é um *data frame* cujas 6 primeiras linhas são:

**Tabela 5 – Saída parcial da componente `pairs` da lista de saída da função `compare.linkage`**

| PAR | id1 | id2 | LOGR _TIPO | LOGR_TITULO _NOME | NUMERO_MODIFICADOR _COMPLEMENTO |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0,73 | 0 |
| 2 | 1 | 2 | 0 | 0,73 | 0 |
| 3 | 1 | 3 | 0 | 0,73 | 0 |
| 4 | 1 | 4 | 0 | 0,73 | 0 |
| 5 | 1 | 5 | 0 | 0,73 | 0 |
| 6 | 1 | 6 | 0 | 0,73 | 0 |

A dimensão do *data frame* `rpairsPACE.11$pairs` é: 139.812 x 6.

A função `emWeights` da library `RecordLinkage` implementa o Algoritmo EM e estima as probabilidades condicionais das configurações que resultam das comparações. Essa função pressupõe que o resultado da comparação de valores para uma variável é 0 se forem distintos ou 1 se forem iguais. Dessa forma, é preciso fixar um patamar para os valores da medida de similaridade de *Jaro-Winlkler*. Valores da medida de similaridade maiores que esse patamar são tomados iguais a 1 e 0 caso contrário. A *library* `RecordLinkage` utiliza o mesmo patamar para todas as variáveis. Os argumentos da função `emWeights` são:

| | |
|---|---|
| `rpairs` | Pares de registros para os quais serão calculados os escores. Objeto criado pela função `compare.linkage`. |
| `cutoff` | Valor numérico no intervalo [0,1] ou um vetor de mesmo comprimento que o número de atributos nos dados. Valor de patamar para comparação da variável de texto. |
| `store.weights` | Valor lógico. Armazena os pesos na base de dados se recebe valor `TRUE`. |
| `verbose` | Valor lógico. Imprime mensagens de progresso se recebe valor `TRUE`. |
| `...` | Argumentos adicionais que são passados para `mygllm`. |

A saída da função é uma cópia de `rpairs` com os escores anexados. Fixando esse patamar em 0,85, a chamada da função seria:

```
> emWeightsPACE.11<-emWeights(rpairsPACE.11,cutoff=0.85)
```

O objeto `emWeightsPACE.11` é uma lista tendo como componentes:

```
[1] "data1"    "data2"    "pairs"    "frequencies" "type"
[6] "M"      "U"      "W"      "Wdata"
```

Disporemos os componentes `M, U e W` dessa lista em uma tabela. Eles são, respectivamente, as probabilidades condicionais das configurações de comparação observadas dado $M$ , $U$ e os escores. Na comparação das 3 variáveis, temos 8 configurações possíveis:

**Tabela 6 – Probabilidades condicionais das configurações e respectivos escores**

| Conf | M | U | Escores |
|---|---|---|---|
| 1 | 3,97E-009 | 4,15E-001 | -26,63 |
| 2 | 1,93E-011 | 4,61E-004 | -24,5 |
| 3 | 1,68E-005 | 1,52E-002 | -9,81 |
| 4 | 8,21E-008 | 1,68E-005 | -7,67 |
| 5 | 2,35E-004 | 5,48E+005 | -11,18 |
| 6 | 1,15E-006 | 6,08E-004 | -9,04 |
| 7 | 9,95E-001 | 2,00E-002 | 5,63 |
| 8 | 4,86E-003 | 2,22E-005 | 7,77 |

A partir dos escores, fixando limites para os erros de tipo I e tipo II, podemos classificar os pares em 3 grupos. Para isso, usamos a função `emClassify`:

```
> paresPACE<-emClassify(emWeightsPACE.11,my=.02,ny=.05)
```

O objeto gerado `paresPACE` é uma lista com os seguintes componentes:

```
 [1] "data1" "data2" "pairs" "frequencies" "type"
 [6] "M"   "U"   "W"   "Wdata"    "prediction"
```

A seguir apresentamos uma tabela com os resultados obtidos do componente `paresPACE$prediction`:

**Tabela 8- Tabela de frequências das decisões de pareamento**

| Não pares | Sem classificação | Pares |
|---|---|---|
| 90.214 | 49.368 | 230 |

A partir da tabela vemos que que foram identificados 90.214 pares incorretos e 230 pares corretos e os restantes 49.368 são casos onde não foi possível classificar.

É possível haver entre os 230 pares corretos obtidos algum registro da PA ou do Censo que apareça em mais de um par. Em geral, para $m$ registros da PA podem corresponder $n$ registros do Censo, e nesse caso dizemos que houve uma associação $m:n$ . No caso da Pesquisa de Avaliação, deseja-se obter uma associação $1:1$ .

A solução do problema é obtida pela maximização da soma dos escores dos pares obtidos, sujeito à condição de que cada registro do arquivo da PA só seja pareado com um registro do arquivo do Censo . Esse problema de otimização pode ser expresso com um problema de programação linear e solucionado por meio do Algoritmo Simplex, utilizando-se a library `lpSolve` do R. Esse método mostrou-se impraticável, para os tamanhos usuais de

arquivos a serem pareados, por ser atingido o limite de memória do R. A solução final do problema foi sugerida por Brito e Montenegro (2009) . A função `solve_LSAP` da *library* `clue` soluciona o problema de atribuição de soma linear usando o Algoritmo Húngaro. Essa solução do problema mostrou-se mais eficiente do que a obtida usando a library `lpSolve`.

O primeiro passo na implementação no uso da função `solve_LSAP` é montar uma matriz com os escores obtidos:

```
> nA<-nrow(PA354390705000135)
> nB<-nrow(CE354390705000135)
> mat_escores<-matrix(paresPACE$Wdata,nA,nB,byrow=TRUE)
```

A célula $(i, j)$ da matriz `mat_escores` corresponde à linha $i$ do *data frame* da PA e à coluna $j$ do *data frame* do Censo. Em seguida, substituímos por zero as células da matriz que correspondem aos valores não pareados no primeiro estágio:

```
> mat_escores[mat_pred!="L"]<-0
```

Adicionamos um bloco de zeros para tornar a matriz quadrada( nA>nB):

```
> mat_0<-matrix(0,nA,nA-nB)
> mat_escores<-cbind(mat_escores,mat_0)
```

A matriz `mat_escores` tem dimensão final $nA \times nA$ e contém os escores nas células correspondentes aos registros pareados e 0 nas outras células.

A função `solve_LSAP` da *library* `clue` implementa a metodologia de otimização descrita para obter a redução 1:1 dos pares.

```
> library(clue)
> result<-solve_LSAP(mat_escores, maximum = TRUE)
```

A partir do objeto `result`, criamos a Tabela 9.

**Tabela 9: Resultado parcial da associação 1:1**

| Escore | id | ULT_NOME_RESP | LOGR_TIPO | LOGR_TITULO_NOME | NUMERO_MODIFICADOR_ COMPLEMENTO |
|---|---|---|---|---|---|
| 7.774056 | 1 | MARQUES | RUA | M NOVE | 731 |
| | 311 | MARQUES | RUA | M NOVE | 731 |
| 7.774056 | 2 | MARCUCI | RUA | M DEZ | 626 |
| | 45 | MARCUCI | RUA | M DEZ | 626 |
| 7.774056 | 3 | AZEVEDO | RUA | M NOVE | 305 |
| | 281 | AZEVEDO | RUA | M NOVE | 305 |
| 7.774056 | 5 | CORREA | RUA | M ONZE | 496 |
| | 229 | CORREA | RUA | M ONZE | 496 |

Essa tabela apresenta as primeiras linhas da matriz de saída. Os registros pareados são apresentados em linhas consecutivas, o primeiro sendo da PA. Os pares apresentados têm escore máximo, todos com vetor $(1,1,1,1)$ de comparação de variáveis.

# 3- Execução de Scripts do R

Todos os *scripts* do R utilizados de deduplicação e pareamento de domicílios e pessoas foram rodados por meio do comando `Rscript`. Para isto foi utilizado o seguinte arquivo em formato .bat:

- Para domicílios:

```
echo %1
cd\PA\Programas R
Rscript dedup.r %1 >> C:/PA/RELATORIOS/R/DEDUPD%1.log 2>>&1
Rscript pareia_prob.r %1 >> C:/PA/RELATORIOS/R/PARESD%1.log 2>>&1
cd\PA\Programas
```

- Para pessoas:

```
echo %1
cd\PA\Programas R
Rscript dedup_pess.r %1 >> C:/PA/RELATORIOS/R/DEDUPP%1.log 2>>&1
Rscript pareia_pess.r %1 >> C:/PA/RELATORIOS/R/PARESP%1.log 2>>&1
cd\PA\Programas
```

O comando na primeira linha permite a entrada no *script* do R de argumentos externos. Nessa aplicação, era transmitido o código do setor a ser pareado. Na segunda linha, especifica-se o diretório de trabalho, onde estão contidos todos os *scripts* do R. Na terceira linha, o comando `Rscript` ativa o R, o argumento lido na primeira linha é transmitido para o *script* `dedup.r` (deduplicação) e ele é executado. Dentro do *script* `dedup.r` há um comando:

```
> id.set.urb<-commandArgs(TRUE)
```

que cria um vetor de caracteres contendo o conteúdo de `%1`. Ainda na linha de número 3, o log da execução do *script* no R é direcionado ao arquivo de nome obtido pela concatenação DEDUP e o código do setor, no diretório indicado. A quarta linha, executa o pareamento, usando o *script* do R `pareia.prob.r`.

Resultados do pareamento e relatórios foram impressos em arquivos definidos pelo sistema planejado pela Diretoria de Informática (DI), especificados nos *scripts* do R.

# 4-Deduplicação de Domicílios

O objetivo da deduplicação de um arquivo é identificar os registros que representam a mesma entidade. Na deduplicação de arquivos tanto da PA como do Censo foi utilizada a função `deduplic` da *library* `RecordLinkage` do R. As variáveis utilizadas na comparação de registros foram:

`TIPO` - Tipo de logradouro;

`TIT_NOME` - Título e Nome concatenados;

`NUM_MOD_COMP` - Número, modificador e complementos concatenados;

`V9004` - Primeiro nome da pessoa responsável pelo domicílio;

`V9006` - Último nome da pessoa responsável pelo domicílio.

As similaridades entre os valores das variáveis: `TIT_NOME`, `V9004` e `V9006` foram medidas por meio do critério de Jaro-Winkler com valores de patamar, respectivamente, 0,85; 0,90 e 0,90. Ou seja, campos com valores do critério de Jaro-Winkler acima do patamar correspondente foram considerados coincidentes.

Antes de aplicar a função `deduplic` ao arquivo, foram feitas as seguintes modificações:

`PTO_REF` - valores faltantes receberam o código "`S/REF`";
`V4001` - foram considerados somente registros com `V4001=1`.

Os argumentos da função `deduplic` são:

`dados`: arquivos de dados;
`urb`: `TRUE` se o setor for urbano;
`strcomp`: variáveis a serem comparadas pelo critério de Jaro-Winkler;
`patamar`: vetor de valores de patamar para as variáveis definidas em `strcomp`.

A saída da função `deduplic` é um vetor contendo nas posições consecutivas as linhas correspondentes aos registros duplicados. Por exemplo, o vetor de saída (1,39,2,28,7,12,...) indicaria a duplicação dos registros nas linhas (1,39); (2,28); (7,12), etc.

É importante ressaltar que não houve exclusão de registros considerados duplicados como resultado da deduplicação.

Para a deduplicação de pessoas, foram utilizadas as variáveis: primeiro nome, último nome, relação com responsável, sexo e idade, sendo aplicada a mesma metodologia da deduplicação de domicílios.

# 5-Pareamento de Domicílios

Na aplicação do pareamento probabilístico na Pesquisa de Avaliação, o esforço principal foi concentrado no pareamento de domicílios. O Anexo 1 contém o *script* do R com os comandos utilizados nessa aplicação. Inicialmente foram carregadas as *libraries* utilizadas:

- `RecordLinkage` – que contém funções para implementar os passos do processo do pareamento probabilístico descrito na Seção 1;
- `Clue` – que implementa a associação 1:1, depois de obtido o pareamento m:n pela utilização das funções da library RecordLinkage;
- `sendmailR` – utilizada para emitir relatório sucinto via internet do resultado do pareamento de pares de setores.

Em seguida, foram carregadas as funções auxiliares utilizadas no processo de pareamento e não contidas nas *libraries* citadas. Essas funções foram programadas para atender requisitos específicos do problema de pareamento no caso da Pesquisa de Avaliação.

As variáveis usadas na comparação de registros são:

- `TIPO` - Tipo de logradouro;
- `TIT_NOME` - Título e Nome concatenados;
- `NUMERO_VAL` - Parte de valor do número;
- `COMPLEMENTO` - Complemento de endereço;
- `V9004` - Primeiro nome da pessoa responsável pelo domicílio;
- `V9006` - Último nome da pessoa responsável pelo domicílio;
- `V9008` - Total de homens;
- `V9009` - Total de mulheres.

As variáveis `"TIT_NOME","V9004" e "V9006"` foram comparadas pelo critério de similaridade de Jaro-Winkler com respectivos patamares: 0,85, 0,90 e 0,90. Para as outras variáveis, foi utilizado o critério de coincidência exata.

A partir do número do setor informado pelo sistema, os arquivos correspondentes da PA e do Censo são lidos em diretórios, estabelecidos previamente, para setores urbanos, rurais ou aglomerados subnormais. Inicialmente, a identificação do tipo de setor teria influência no método adotado de pareamento probabilístico. A inclusão do método de seleção de variáveis no procedimento permitiu que fosse adotado um só método para os três tipos de setores. Depois da leitura dos arquivos algumas variáveis foram recodificadas:

- Valor faltante na variável `PTO_REF` (NA) foi substituído por "`S/REF`";

- Foram filtrados os registros com valores `VA4000=1`, tanto no arquivo do Censo como no da PA;

Foi criada uma nova variável `pares.aux`, obtida pela concatenação do primeiro nome do responsável pelo domicílio e do seu cônjuge. A ordem da concatenação foi determinada de modo a maximizar a similaridade dos strings obtidos, medido pelo critério de Jaro-Winkler.

Utilizando a função `varexcl`, foram excluídas variáveis com pouco poder de discriminação. Por exemplo, as variáveis `TIPO` e `TIT_NOME` assumiam em vários setores o mesmo valor para todos os domicílios. Nessas situações, essas variáveis sempre concordavam ou sempre discordavam, implicando o aparecimento de muitas células com 0 na tabela de frequências das configurações de comparação das variáveis. Nos casos extremos, ocorreram problemas na aplicação do Algoritmo EM. Para evitar o problema, foi feita a exclusão de variáveis com pequeno poder de discriminação.

A função `pareia.em`, no Anexo 6- Funções Auxiliares, executa todos os passos do pareamento probabilístico usado na Pesquisa de Avaliação. Nessa função são chamadas as seguintes funções da library `RecordLinkage`:

- `compare.linkage` - compara cada registro do arquivo da PA com cada registro do arquivo do Censo;
- `emWeights1`- modificação da função `emWeights`, que permite usar valores distintos para o patamar da medida de similaridade de Jaro-Winkler para diferentes variáveis;
- `emClassify` - estima os escores utilizando o Agoritmo EM, para níveis fixados dos erros;
- `solve_LSAP` - função da *library* `clue` do R que executa a associação 1:1.

A função `pareia.em`, além de executar o pareamento probabilístico 1:1, implementa alguns critérios determinísticos adicionais de concordância das variáveis, que têm como objetivo diminuir a probabilidade de falsos positivos.

A seguir listamos as condições utilizadas:

1. Número de concordâncias maior ou igual a cinco (quaisquer variáveis); ou
2. Número de concordâncias maior ou igual a 4 incluindo obrigatoriamente a concordância no primeiro nome do responsável concatenado com o primeiro nome do cônjuge; ou
3. Concordância nas variáveis nome do responsável concatenado com o cônjuge, sobrenome e número; ou
4. Concordância nas variáveis primeiro nome do responsável concatenado com o do cônjuge, sobrenome do responsável e complemento.

Essas condições correspondem ao valor do argumento `criterio=1` da função `pareia.em`.

# 6-Pareamento de Pessoas

O pareamento de pessoas utilizou como blocos os domicílios pareados e que pertenciam à amostra da PA. Foram aplicados procedimentos similares aos utilizados no pareamento de domicílios.

Foram calculadas medidas de similaridade para as comparações das seguintes variáveis:

1 - primeiro nome da pessoa;
2 - último sobrenome da pessoa; e
3 - idade da pessoa.

Diferentemente do pareamento de domicílios, os pesos de concordância foram pré-fixados ou seja, para cada vetor de comparação foi associado um peso.

**Tabela 10 – Pesos utilizados no pareamento de pessoas**

| Concordância nas variáveis | Peso |
|---|---|
| Idade, Primeiro e Último nome | 1 |
| Primeiro nome e Idade | 0,8 |
| Primeiro e Último nome | 0,7 |

Qualquer concordância diferente da tabela possui peso associado igual a zero.

Foi aplicada a técnica de redução 1:1 e os os pares que possuíam pesos diferentes de zero foram considerados verdadeiros. O *script* utilizado no pareamento de pessoas está descrito no Anexo 2.

# 7- Pareamento em Setores Contíguos

Para os domicílios e pessoas listados na PA que não tiveram pares corretos encontrados dentro do setor correspondente no Censo, foi feita a busca do par correto dentre os domicílios e pessoas listados na coleta do Censo nos setores contíguos.

Todos os procedimentos aplicados no pareamento probabilístico foram novamente aplicados, utilizando-se os setores contíguos em vez do setor correspondente.

Os critérios adotados na seleção dos setores contíguos foram definidos e aplicados pela Diretoria de Geociências, que forneceu a listagem dos mesmos. O *script* utilizado no pareamento de setores contíguos é apresentado no Anexo 5.

# Referências Bibliográficas


Andreas Borg <reclinkmainz@googlemail.com> and Murat Sariyar <reclinkmainz@google mail.com> (2011). RecordLinkage: Record Linkage in R. R package version 0.3-5. URL http://CRAN.R-project.org/package=RecordLinkage.

Brito, J.A. e Montenegro, F. (2009). Associação 1:1 como um problema de atribuição. Seminário. ENCE.

Fellegi, I.P. e Sunter , A.B. (1969). A Theory for Record Linkage, *Journal of the American Statistical Association*, Vol. **64**, No. 328.

Kurt Hornik (2011). clue: Cluster ensembles. R package version 0.3-42. URL http://CRAN.R-project.org/package=clue.

Michel Berkelaar and others (2011). lpSolve: Interface to Lp_solve v. 5.5 to solve linear/integer programs. R package version 5.6.6. URL http://CRAN.R-project.org/package= lpSolve.

Olaf Mersmann <olafm@datensplitter.net> (2011). sendmailR: send email using R. R package version 1.1-1. http://CRAN.R-project.org/package=sendmailR.

R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Silva, A.D. et all. (2011). Pesquisa de Avaliação da Cobertura da Coleta do Censo 2010. Relatório Final.

# Anexos

## Anexo 1- Pareamento de Domicílios

```
#############################################################################
###                            IBGE/DPE/COMEQ                             ###
###         Pareamento de domicílios e pessoas do CENSO e da PA           ###
###                            Setembro/2010                              ###
#############################################################################
###              PAREAMENTO PROBABILÍSTICO (EM) - DOMICÍLIOS              ###
#############################################################################
### Autores: Djalma Pessoa                                                ###
###           Fábio Figueiredo                                            ###
###           Vinicius Xavier                                             ###
#############################################################################

tempo<-Sys.time()

################
### Funções ###
################

# Carrega libraries e funções:

library(RecordLinkage)
library(clue)
library(sendmailR)
source("F:/PA/PROGRAMAS R/PesqAval.r")

#######################################
### Variáveis usadas no pareamento ###
#######################################

var.nomes<-c("TIPO","TIT_NOME","NUMERO_VAL","COMPLEMENTO","V9004","V9006",
"V9008","V9009")

# Variávels comparadas pelo critério de Jaro-Winkler:
jaro.num<-which(var.nomes%in%c("TIT_NOME","V9004","V9006"))

# Patamar para o critério de Jaro-Winkler
patamar.jaro<-c(1,0.85,1,1,0.90,0.90,1,1)

# Saída de mensagens de erro do R

sink(paste("F:/PA/RELATORIOS/ERROS/HISTORICOD.txt"),append=TRUE)
cat("\n",substring(tempo,1,19),"\n",sep="")
cat("Setor:","\n",sep="")
sink()
tempo<-gsub(":","_",tempo)

#################
### Preâmbulo ###
#################

#Localização do setor
nomes.arq.urb<-list.files("F:/PA/DOMICILIOS/URBANO")
nomes.arq.rur<-list.files("F:/PA/DOMICILIOS/RURAL")
nomes.arq.sub<-list.files("F:/PA/DOMICILIOS/AGL_SUB_NOR")
```

```
# Recebe identificação do setor
id.set<-commandArgs(TRUE)

if(!id.set%in%substring(c(nomes.arq.urb,nomes.arq.sub,nomes.arq.rur),2,16))
print(paste("O setor",id.set,"não foi encontrado na pasta de entrada.")
)else{
  tipo<-"urb"
  nomes.arq<-id.set
  if(id.set%in%substring(nomes.arq.sub,2,16))tipo<-"sub"
  if(id.set%in%substring(nomes.arq.rur,2,16))tipo<-"rur"
  path<- "URBANO"
  urb<-TRUE
  if(tipo=="sub"){
   path<-"AGL_SUB_NOR"
   urb<-TRUE
  }
  if(tipo=="rur"){
   path<-"RURAL"
   urb<-FALSE
  }

  setwd(paste("F:/PA/DOMICILIOS/",path,sep=""))

  list.CE<-paste("C",id.set,sep="")
  list.PA<-paste("P",id.set,sep="")

  #################
  ### Execução ###
  #################

  resumo<-matrix(NA,length(id.set),6)
  dimnames(resumo)<-list(id.set,c("n_CE","n_PA","n_Pares","%_Pares",
   "n_RES_CE","n_RES_PA"))
  list.res<-list()
  list.id<-list()
  nreg<-matrix(NA,nrow=length(id.set),ncol=2)
  print(id.set)

# Leitura dos arquivos do Censo e da PA
  assign("dados.CE0",read.table(paste(list.CE,"txt",sep="."),
   header=TRUE,colClasses="character",sep=";",na.strings=""))
  assign("dados.PA0",read.table(paste(list.PA,"txt",sep="."),
   header=TRUE,colClasses="character",sep=";",na.strings=""))

# Recodificação e filtragem dos dados:

  if(nrow(dados.CE0)>0 & nrow(dados.PA0)>0){
   dados.CE0[is.na(dados.CE0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
   dados.PA0[is.na(dados.PA0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
   dados.CE0[,"V4001"]<-as.integer(dados.CE0[,"V4001"])
   dados.PA0[,"V4001"]<-as.integer(dados.PA0[,"V4001"])
   dados.CE0<-subset(dados.CE0,V4001==1)
   dados.PA0<-subset(dados.PA0,V4001==1)
   dados1<-dados.CE0[,var.nomes]
   dados2<-dados.PA0[,var.nomes]
   nreg[1,]<-c(nrow(dados1),nrow(dados2))


## Concatena primeiros nomes dos membros do casal e
## usa a ordem de concatenação que fornece valor máximo
## do critério de Jaro-Winkler
```

```
   nom.conjCE<-dados.CE0[,"V9007"]
   nom.conjPA<-dados.PA0[,"V9007"]
   CE.V9007.P<-unlist(lapply(strsplit(nom.conjCE," "),function(t)t[1]))
   PA.V9007.P<- unlist(lapply(strsplit(nom.conjPA," "),function(t)t[1]))
   CE.V9007.U<-unlist(lapply(strsplit(nom.conjCE," "),function(t)t[
    length(t)]))
   PA.V9007.U<-unlist(lapply(strsplit(nom.conjPA," "),function(t)t[
    length(t)]))
   CE.VAUX1<-paste(dados1$V9004,CE.V9007.P,sep="")
   CE.VAUX2<-paste(dados1$V9004,CE.V9007.P,sep="")
   PA.VAUX1<-paste(dados2$V9004,PA.V9007.P,sep="")
   PA.VAUX2<-paste(PA.V9007.P,dados2$V9004,sep="")
   AUX.CE<-data.frame(VAUX1= CE.VAUX1, VAUX2= CE.VAUX2)
   AUX.PA<-data.frame(VAUX1= PA.VAUX1, VAUX2= PA.VAUX2)
   pares.aux<-compare.linkage(AUX.CE,AUX.PA,strcmp=c(1,2),strcmpfun=
    jarowinkler)$pairs
   conj.jaro<-pmax(pares.aux$VAUX1,pares.aux$VAUX2)

   ### Excluir variáveis de modo a evitar problemas no uso do EM ###

   var.num<-varexcl(dados1,dados2,jaro.num,patamar.jaro)
   var.nomes1<-var.nomes[var.num]
   jaro.num<-which(var.nomes1%in%c("TIT_NOME","V9004","V9006"))
   patamar.jaro=patamar.jaro[var.num]
   dados1<-dados.CE0[,var.nomes1]
   dados2<-dados.PA0[,var.nomes1]

   ### Relatório em formato txt ###
   dados.CE<-dados.CE0[,var.nomes]
   dados.PA<-dados.PA0[,var.nomes]
   dif.nom.CE<-lapply(dados.CE,unique)
   tab.nom.CE<-lapply(dados.CE,table)
   dif.nom.PA<-lapply(dados.PA,unique)
   tab.nom.PA<-lapply(dados.PA,table)
   sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD", id.set,".txt",sep=""),
    append=FALSE)
   cat("\nRelatório do Setor:",id.set,"\n")
   cat("\nNomes distintos da variável no Censo:","\n")
   print(tab.nom.CE)
   cat("\nNomes distintos da variável na PA:","\n")
   print(tab.nom.PA)
   cat("\nVariáveis utilizadas:","\n")
   print(var.nomes1)
   sink()


### Pareamento ###

   res.pares<-pareia.em(dados1=dados1,dados2=dados2,urb=urb,
   strcomp=jaro.num,patamar=patamar.jaro,criterio=1,conj.aux=conj.jaro)
   Pares.em<-res.pares[[1]]

   ### Se fez pares -> Redução 1 a 1, senão -> Relatório ###
   if(nrow(Pares.em)>0){
     ID_CE<-Pares.em[,"id1"]
     ID_PA<-Pares.em[,"id2"]
     var.com<-intersect(names(dados.PA0),names(dados.CE0))
     pares.finais<-cbind(Escore=Pares.em[,"Escore"],Conc=Pares.em[,"Conc"],
      ID_CE,dados.CE0[ID_CE,var.com],Conc=Pares.em[,"Conc"],ID_PA,dados.PA0
      [ID_PA,var.com])
     pares.finais<-pares.finais[order(pares.finais[,"Escore"],decreasing=
      TRUE),]
     list.id[[1]]<-paste(pares.finais[,5],pares.finais[,30],sep=".")
```

```
   duplas<-duplica.linhas(pares.finais,c("Escore","Conc","ID",var.com))
   list.res[[1]]<-list(duplas,res.pares[[2]],res.pares[[3]])

   ### Obtenção dos resíduos: registros não-pareados ###
   res.PA<-setdiff(1:nrow(dados.PA0),pares.finais[,"ID_PA"])
   res.CE<-setdiff(1:nrow(dados.CE0),pares.finais[,"ID_CE"])
   resA<-cbind(ID_CE=res.CE,dados.CE0[res.CE,])
   resB<-cbind(ID_PA=res.PA,dados.PA0[res.PA,])
   duplas<-subset(duplas,select=V9001)

   ### Exportando os pares no formato txt ###
   write.table(duplas,file=paste("F:/PA/PARESD/PARD",id.set,".txt",
    sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE,col.names=FALSE)

   ### Resumo do pareamento ###
   resumo[1,1]<-nrow(dados1)
   resumo[1,2]<-nrow(dados2)
   resumo[1,3]<-nrow(pares.finais)
   resumo[1,4]<-round(100*resumo[1,3]/min(resumo[1,1],resumo[1,2]),1)
   resumo[1,5]<-nrow(resA)
   resumo[1,6]<-nrow(resB)
 }else{
   resumo[1,1]<-nrow(dados1)
   resumo[1,2]<-nrow(dados2)
   resumo[1,3]<-0
   resumo[1,4]<-0
   resumo[1,5]<-nrow(dados.CE0)
   resumo[1,6]<-nrow(dados.PA0)
   resA<-dados.CE0
   resB<-dados.PA0
 }

 ### Relatório em formato txt ###
 intercPACE<-integer(length(var.nomes))
 mat.crit<-matr.crit<-matrix(NA,3,length(var.nomes))
 dimnames(mat.crit)<-list(c("dif_CE","dif_PA","dif_int"),var.nomes)
 for(j in 1:length(var.nomes))intercPACE[j]<-length(intersect(dif.nom.CE[
  [j]],dif.nom.PA[[j]]))
 mat.crit[1,]<-sapply(dif.nom.CE,length)
 mat.crit[2,]<-sapply(dif.nom.PA,length)
 mat.crit[3,]<-intercPACE
 sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD",id.set,".txt",sep=""),
  append=TRUE)
 cat("\nFrequências das interseções para Censo e PA:","\n")
 print(mat.crit)
 cat("\nResumo:","\n")
 print(resumo[1,])
 sink()
 sink("F:/PA/RELATORIOS/ERROS/HISTORICOD.txt",append=TRUE)
 cat(id.set,".....................ok","\n",sep="")
 sink()
}else{
 sink("F:/PA/RELATORIOS/ERROS/HISTORICOD.txt",append=TRUE)
 cat(id.set,".........Não processado","\n",sep="")
 sink()
}

resumo<-cbind(resumo,AVISO=aprov(resumo))

### Copiar arquivos para a pasta de PROCESSADOS e
### e remover da pasta de origem

file.copy(from=paste("F:/PA/DOMICILIOS/",path,"/C",id.set,".txt",
```

```
   sep=""),to=paste("F:/PA/PROCESSADOSD/",path,"/C",id.set,".txt",sep=""),
   overwrite=TRUE)
  file.remove(paste("F:/PA/DOMICILIOS/",path,"/C",id.set,".txt",sep=""))
  file.copy(from=paste("F:/PA/DOMICILIOS/",path,"/P",id.set,".txt",
   sep=""),to=paste("F:/PA/PROCESSADOSD/",path,"/P",id.set,".txt",sep=""),
   overwrite=TRUE)
  file.remove(paste("F:/PA/DOMICILIOS/",path,"/P",id.set,".txt",sep=""))
  msg<-paste(" SETOR ",path,dimnames(resumo)[[2]][1],dimnames(resumo)[[2]
   ][2],dimnames(resumo)[[2]][3],dimnames(resumo)[[2]][4],dimnames(resumo
   )[[2]][5],dimnames(resumo)[[2]][6],"  ",dimnames(resumo)[[2]][7],
   sep=" ")

  ### Gerar arquivo de resumo do lote ###
  write.table(resumo,paste("F:/PA/RELATORIOS/ERROS/LOTE/",tempo,".txt",
   sep=""),quote=FALSE)
  ### Enviar e-mail com o resumo do lote ###
  from<-"R"
  to<-"PA2010@ibge.gov.br"
  subject<-paste("DOMICÍLIOS: " ,row.names(resumo)[1],sep="")
  for(i in 1:nrow(resumo)){
   msg<-paste(msg,"\n",formata(set=row.names(resumo)[i], x=resumo[i,]))
  }
  mail<-sendmail(from,to,subject,msg,control=list(smtpServer="172.31.0.
   107"))
}

## Utilização de patamar menor para auxiliar no pareamento assistido
## Método 3

if(id.set%in%substring(c(nomes.arq.urb,nomes.arq.sub,nomes.arq.rur),2,16)){
  patamar.jaro3<-patamar.jaro
  res.pares3<-pareia.em(dados1=dados1,dados2=dados2,urb=urb,
  strcomp=jaro.num,patamar=patamar.jaro3,criterio=2,conj.aux=conj.jaro)
  Pares.em3<-res.pares3[[1]]
  list.res3<-list()
  list.id3<-list()

  ### Se fez pares -> Redução 1 a 1, senão -> Relatório ###
  if(nrow(Pares.em3)>0){
   ID_CE3<-Pares.em3[,"id1"]
   ID_PA3<-Pares.em3[,"id2"]
   var.com<-intersect(names(dados.PA0),names(dados.CE0))
   pares.finais3<-cbind(Escore=Pares.em3[,"Escore"],Conc=Pares.em3[,
    "Conc"],ID_CE3,
   dados.CE0[ID_CE3,var.com],Conc=Pares.em3[,"Conc"],ID_PA3,dados.PA0[ID_
    PA3,var.com])
   pares.finais3<-pares.finais3[order(pares.finais3[,"Escore"],decreasing=
    TRUE),]
   list.id3[[1]]<-paste(pares.finais3[,5],pares.finais3[,30],sep=".")
   duplas3<-duplica.linhas(pares.finais3,c("Escore","Conc","ID",var.com))
   list.res3[[1]]<-list(duplas3,res.pares[[2]],res.pares[[3]])
   duplas3<-subset(duplas3,select=V9001)
   ### Exportando os pares no formato txt ###
   write.table(duplas3,file=paste("F:/PA/PARESD2/PARD", id.set,".txt",
sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
```

## Anexo 2 – Pareamento de Pessoas

```
#############################################################################
###                            IBGE/DPE/COMEQ                             ###
###         Pareamento de domicílios e pessoas do CENSO e da PA           ###
###                            Setembro/2010                              ###
#############################################################################
###                  PAREAMENTO DETERMINÍSTICO - PESSOAS                  ###
#############################################################################
### Autores: Djalma Pessoa                                                ###
###      Fábio Figueiredo                                                 ###
###       Vinicius Xavier                                                 ###
#############################################################################

tempo<-Sys.time()

################
### Funções ###
################
library(RecordLinkage)
library(clue)
source("F:/PA/PROGRAMAS R/PesqAval.r")

###############################
### Aviso de erro do setor ###
###############################
aprov<-function(x){
 if(!is.na(x[4])&(as.numeric(x[4])/as.numeric(x[3]))>.10)
 "Aprovado"else
 "Reprovado"
}

##################
### Variáveis ###
##################
var.nomes.pes<-c("V9019","V9021","V6033")
jaro.num.pes1<-c("V9019","V9021")
patamar.jaro.pes<-c(0.85,0.9,1)
sink(paste("F:/PA/RELATORIOS/ERROS/HISTORICOP.txt"),append=TRUE)
 cat("\n",substring(tempo,1,19),"\n",sep="")
 cat("Setor:","\n",sep="")
sink()
tempo<-gsub(":","_",tempo)

#################
### Preâmbulo ###
#################
setwd("F:/PA/PESSOAS")
nomes.arq.pessoa<-list.files()
nomes.arq.pessoa<-nomes.arq.pessoa[grep("txt",nomes.arq.pessoa)]
id.set.pessoa<-commandArgs(TRUE)
print(id.set.pessoa)
list.CE.pessoa<-paste("C",id.set.pessoa,sep="")
list.PA.pessoa<-paste("P",id.set.pessoa,sep="")
id.set.pessoa<-ifelse(id.set.pessoa%in%substring(nomes.arq.pessoa,2,16),
id.set.pessoa,character(0))


################
### Execução ###
################
```

```
roda<-function(id.set.pessoa,list.CE.pessoa,list.PA.pessoa, var.nomes.pes,
jaro.num.pes1,patamar.jaro.pes,tempo){

 if(is.na(id.set.pessoa))stop("O setor não foi encontrado na pasta de
  entrada")
 resumo.pessoa<-matrix(NA,length(id.set.pessoa),7)
 dimnames(resumo.pessoa)<-list(id.set.pessoa,c("n_CE","n_PA","n_max_Pares",
  "n_Pares","%_Pares","n_RES_CE","n_RES_PA"))
 assign("pessoa.CE",read.table(paste(list.CE.pessoa,"txt",sep="."),header=
  TRUE,colClasses="character",sep=";",na.strings=""))
 assign("pessoa.PA",read.table(paste(list.PA.pessoa,"txt",sep="."),header=
  TRUE,colClasses="character",sep=";",na.strings=""))
 if (nrow(pessoa.CE)>0&nrow(pessoa.PA)>0){
  var.com<-intersect(names(pessoa.CE),names(pessoa.PA))
  if(sum(var.nomes.pes%in%var.com)<3) stop("Variáveis usadas na comparação
não encontradas")
  var.excl<-which(!var.com%in%var.nomes.pes)
  jaro.num.pes<-which(var.com%in%jaro.num.pes1)
  dados1<-pessoa.CE[,var.com]
  dados2<-pessoa.PA[,var.com]
  var.bloco<-which(names(dados1)=="V9012")
  rpairs<-compare.linkage(dados1,dados2,blockfld=var.bloco,exclude=
   var.excl,
  strcmp=jaro.num.pes,strcmpfun=jarowinkler)
  comp.dados<-rpairs$pairs
  idade1<-as.numeric(dados1[comp.dados[,"id1"],"V6033"])
  idade2<-as.numeric(dados2[comp.dados[,"id2"],"V6033"])
  escores.idade<-peso.idade(idade1,idade2)
  escores.idade1<-rep(0,length(escores.idade))
  escores.idade1[escores.idade>=0.1]<-1
  comp.dados<-subset(comp.dados,select=-is_match)
  comp.dados<-as.matrix(comp.dados)
  ind.na<-is.na(comp.dados)
  comp.dados[ind.na]<-0
  #comp.dados1<-comp.dados[,-c(1,2)]
  temporario<-subset(comp.dados,select=-id1)
  comp.dados1<-subset(temporario,select=-id2)
  #comp.dados1<-as.matrix(comp.dados1)
  for(j in 1:3){
   1:ncol(comp.dados1)
   comp.dados1[,j]<-(comp.dados1[,j]>=patamar.jaro.pes[j])*1
  }
  comp.dados1[,"V6033"]<-escores.idade1
  comp.dados[,-c(1,2)]<-comp.dados1
  escores_det<-apply(comp.dados1,1,esc.pess)
  comp.dados<-cbind(comp.dados,escores_det)
  comp.dados<-as.data.frame(comp.dados)
  PARID<-dados1[comp.dados[,"id1"],"V9012"]
  list.dom<-split(comp.dados,PARID)
  list.pares.pes<-sapply(list.dom,assoc11, simplify = FALSE)
  mat.final<-list.pares.pes[[1]]
  if(length(list.pares.pes)>1)for(j in 2:length(list.pares.pes)
   )mat.final<-rbind(mat.final,list.pares.pes[[j]])
  mat.final<-mat.final[mat.final[,"id1"]>0,,drop=FALSE]
  if(nrow(mat.final)==0)stop("Nenhum par formado")
  pares.pessoa<-cbind(dados1[mat.final[,"id1"],"V9011"],dados2[mat.final[,
   "id2"],"V9011"])
  duplas.pes<-cbind(1,pares.pessoa)
  duplas.pes<-duplica.linhas(duplas.pes,c("UM","V9011"))
  duplas.pes<-subset(duplas.pes,select=-UM)
  write.table(duplas.pes,file=paste("F:/PA/PARESP/PARP",
  id.set.pessoa,".txt",sep=""),row.names=FALSE,col.names=FALSE,quote=FALSE)
  resumo.pessoa[1,1]<-length(unique(pessoa.CE[,"V9011"]))
```

```
 resumo.pessoa[1,2]<-length(unique(pessoa.PA[,"V9011"]))
 resumo.pessoa[1,3]<-sum(sapply(list.dom,n.max.pares))
 resumo.pessoa[1,4]<-nrow(pares.pessoa)
 resumo.pessoa[1,5]<-round(100*resumo.pessoa[1,4]/resumo.pessoa[1,3],1)
 resumo.pessoa[1,6]<-resumo.pessoa[1,1]-resumo.pessoa[1,4]
 resumo.pessoa[1,7]<-resumo.pessoa[1,2]-resumo.pessoa[1,4]
 sink("F:/PA/RELATORIOS/ERROS/HISTORICOP.txt",append=TRUE)
  cat(id.set.pessoa,".....................ok","\n",sep="")
 sink()
}else{
 sink("F:/PA/RELATORIOS/ERROS/HISTORICOP.txt",append=TRUE)
 cat(id.set.pessoa,".........Não processado","\n",sep="")
 sink()
}
resumo.pessoa<-cbind(resumo.pessoa,AVISO=aprov(resumo.pessoa))

### Copiar arquivos para a pasta de PROCESSADOS e remover da pasta de
### origem
nomes.arq.pessoa<-nomes.arq.pessoa[grep(id.set.pessoa,nomes.arq.pessoa)]
file.copy(from=paste("F:/PA/PESSOAS/",nomes.arq.pessoa,sep=""),
to=paste("F:/PA/PROCESSADOSP/",nomes.arq.pessoa,sep=""),overwrite=TRUE)
file.remove(paste("F:/PA/PESSOAS/",nomes.arq.pessoa,sep=""))

### Gerar arquivo de resumo do lote ###
write.table(resumo.pessoa,paste("F:/PA/RELATORIOS/ERROS/PESSOAS/",
tempo,".txt",sep=""),quote=FALSE)

### Enviar e-mail com o resumo do lote ###
library(sendmailR)
from<-"R"
to<-"pa2010@ibge.gov.br"
subject<-paste("PESSOAS: ",row.names(resumo.pessoa)[1],sep="")
msg<-paste(" SETOR      ",dimnames(resumo.pessoa)[[2]]
[1],dimnames(resumo.pessoa)[[2]][2],dimnames(resumo.pessoa)[[2]]
[3],dimnames(resumo.pessoa)[[2]][4],dimnames(resumo.pessoa)[[2]]
[5],dimnames(resumo.pessoa)[[2]][6]," ",dimnames(resumo.pessoa)[[2]][7],
"    ",dimnames(resumo.pessoa)[[2]][8])
for (i in 1:nrow(resumo.pessoa)){
 msg<-paste(msg,"\n",formata(set=row.names(resumo.pessoa)[i],x=resumo.
  pessoa[i,]))
}
sendmail(from,to,subject,msg,control=list(smtpServer="172.31.0.107"))
}
roda(id.set.pessoa=id.set.pessoa,list.CE.pessoa=list.CE.pessoa,list.PA.pess
 oa=list.PA.pessoa,var.nomes.pes=var.nomes.pes,jaro.num.pes1=jaro.num.pes1,
 patamar.jaro.pes=patamar.jaro.pes,tempo=tempo)
```

## Anexo 3 – Deduplicação de Domicílios

```
#############################################################################
###                            IBGE/DPE/COMEQ                             ###
###         Pareamento de domicílios e pessoas do CENSO e da PA           ###
###                            Setembro/2010                              ###
#############################################################################
###                       DEDUPLICAÇÃO - DOMICILIOS                       ###
#############################################################################
### Autores: Djalma Pessoa                                                ###
###          Fábio Figueiredo                                             ###
###          Vinicius Xavier                                              ###
#############################################################################

tempo<-Sys.time()

###############
### Funções ###
###############
library(RecordLinkage)
library(clue)
source("F:/PA/PROGRAMAS R/PesqAval.r")

#################
### Variáveis ###
#################
var.nomes<-c("TIPO","TIT_NOME","NUM_MOD_COMP","V9004","V9006")
jaro.num<-which(var.nomes%in%c("TIT_NOME","V9004","V9006"))
patamar.jaro<-c(1,0.85,1,0.90,0.90)

##########################
### Preâmbulo - Urbano ###
##########################
setwd("F:/PA/DOMICILIOS/URBANO")
nomes.arq.urb<-list.files()
nomes.arq.urb<-nomes.arq.urb[grep("txt",nomes.arq.urb)]
id.set.urb<-commandArgs(TRUE)
list.CE.urb<-paste("C",id.set.urb,sep="")
list.PA.urb<-paste("P",id.set.urb,sep="")
id.set.urb<-ifelse(id.set.urb%in%substring(nomes.arq.urb,2,16),id.set.urb,
 character(0))

##########################
### Execução - Urbano ###
##########################
if(!is.na(id.set.urb)){
 print(id.set.urb)
 ### Censo ###
 assign("dados.CE0",read.table(paste(list.CE.urb,"txt",sep="."),header=TRUE
  ,colClasses="character",sep=";",na.strings=""))
 if (nrow(dados.CE0)==0){
  sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD",id.set.urb,".txt",sep=""
   ),append=TRUE)
  cat("\n",paste("Setor ",id.set.urb," do Censo tem zero linhas."),"\n")
  sink()
 }else{
  dados.CE0[is.na(dados.CE0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
  dados.CE0[,"V4001"]<-as.integer(dados.CE0[,"V4001"])
  dados.CE0<-subset(dados.CE0,V4001==1)
  dados1<-dados.CE0[,var.nomes]
  CE.deduplic<-deduplic(dados=dados1,strcomp=jaro.num,patamar=patamar.jaro)
  if(length(CE.deduplic)>1){
```

```
   duplas.CE<-dados.CE0[CE.deduplic,]
   write.table(duplas.CE,file=paste("F:/PA/DEDUPD/DEDUPD_C",
   id.set.urb,".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE)
  }
 }
 ### PA ###
 assign("dados.PA0",read.table(paste(list.PA.urb,"txt",sep="."),
 header=TRUE,colClasses="character",sep=";",na.strings=""))
 if (nrow(dados.PA0)==0){
  sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD",id.set.urb,".txt",sep=""
   ),append=TRUE)
  cat("\n",paste("Setor ",id.set.urb," da PA tem zero linhas."),"\n")
  sink()
 }else{
  dados.PA0[is.na(dados.PA0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
  dados.PA0[,"V4001"]<-as.integer(dados.PA0[,"V4001"])
  dados.PA0<-subset(dados.PA0,V4001==1)
  dados2<-dados.PA0[,var.nomes]
  PA.deduplic<-deduplic(dados=dados2,strcomp=jaro.num,patamar=patamar.jaro)
  if(length(PA.deduplic)>1){
   duplas.PA<-dados.PA0[PA.deduplic,]
   write.table(duplas.PA,file=paste("F:/PA/DEDUPD/DEDUPD_P",
   id.set.urb,".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE)
  }
 }
}

#########################
### Preâmbulo - Rural ###
#########################
setwd("F:/PA/DOMICILIOS/RURAL")
nomes.arq.rur<-list.files()
nomes.arq.rur<-nomes.arq.rur[grep("txt",nomes.arq.rur)]
id.set.rur<-commandArgs(TRUE)
list.CE.rur<-paste("C",id.set.rur,sep="")
list.PA.rur<-paste("P",id.set.rur,sep="")
id.set.rur<-ifelse(id.set.rur%in
%substring(nomes.arq.rur,2,16),id.set.rur,character(0))

########################
### Execução - Rural ###
########################
if(!is.na(id.set.rur)){
 print(id.set.rur)
 ### Censo ###
 assign("dados.CE0",read.table(paste(list.CE.rur,"txt",sep="."),
 header=TRUE,colClasses="character",sep=";",na.strings=""))
 if (nrow(dados.CE0)==0){
  sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD",id.set.rur,".txt",sep=""
   ),append=TRUE)
  cat("\n",paste("Setor ",id.set.rur," do Censo tem zero linhas."),"\n")
  sink()
 }else{
  dados.CE0[is.na(dados.CE0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
  dados.CE0[,"V4001"]<-as.integer(dados.CE0[,"V4001"])
  dados.CE0<-subset(dados.CE0,V4001==1)
  dados1<-dados.CE0[,var.nomes]
  CE.deduplic<-deduplic(dados=dados1,strcomp=jaro.num,patamar=patamar.jaro)
  if(length(CE.deduplic)>1){
   duplas.CE<-dados.CE0[CE.deduplic,]
   write.table(duplas.CE,file=paste("F:/PA/DEDUPD/DEDUPD_C",id.set.rur,
    ".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE)
  }
```

```
 }
 ### PA ###
 assign("dados.PA0",read.table(paste(list.PA.rur,"txt",sep="."),header=TRUE
  ,colClasses="character",sep=";",na.strings=""))
 if (nrow(dados.PA0)==0){
  sink(paste("F:/PA/RELATORIOS/ERROS/SETOR/RELATD",id.set.rur,".txt",sep=""
   ),append=TRUE)
  cat("\n",paste("Setor ",id.set.rur," da PA tem zero linhas."),"\n")
  sink()
 }else{
  dados.PA0[is.na(dados.PA0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
  dados.PA0[,"V4001"]<-as.integer(dados.PA0[,"V4001"])
  dados.PA0<-subset(dados.PA0,V4001==1)
  dados2<-dados.PA0[,var.nomes]
  PA.deduplic<-deduplic(dados=dados2,strcomp=jaro.num,patamar=patamar.jaro)
  if(length(PA.deduplic)>1){
   duplas.PA<-dados.PA0[PA.deduplic,]
   write.table(duplas.PA,file=paste("F:/PA/DEDUPD/DEDUPD_P",id.set.rur,
    ".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE)
  }
 }
}
### Avisar caso não haja arquivo de entrada ###
if(is.na(id.set.urb)&is.na(id.set.rur))print(paste("O setor",commandArgs(
TRUE),"não foi encontrado na pasta de entrada."))
```

## Anexo 4 – Deduplicação de Pessoas

```
##############################################################################
###                              IBGE/DPE/COMEQ                            ###
###        Pareamento de domicílios e pessoas do CENSO e da PA             ###
###                               Setembro/2010                            ###
##############################################################################
###                          DEDUPLICAÇÃO - PESSOAS                        ###
##############################################################################
### Autores: Djalma Pessoa                                                 ###
###          Fábio Figueiredo                                              ###
###          Vinicius Xavier                                               ###
##############################################################################

tempo<-Sys.time()

###############
### Funções ###
###############
library(RecordLinkage)
library(clue)
library(RODBC)
source("F:/PA/PROGRAMAS R/PesqAval.r")

#################
### Variáveis ###
#################

# Primeiro nome # ultimo nome # relação com responsável # sexo #idade #
var.nomes.pes<-c("V9019","V9021","V0502","V0601","V6033")
jaro.num.pes1<-c("V9019","V9021")
patamar.jaro.pes<-c(0.9,0.9,1,1,1)

#################
### Preâmbulo ###
#################
setwd("F:/PA/DEDUPP")
nomes.arq.pessoa<-list.files()
nomes.arq.pessoa<-nomes.arq.pessoa[grep("txt",nomes.arq.pessoa)]
id.set.pessoa<-commandArgs(TRUE)
pesquisa<-c("C","P")
for (i in pesquisa){
 id.set.pesquisa<-paste(i,id.set.pessoa,sep="")
 id.set.pesquisa<-ifelse(id.set.pesquisa%in%substring(nomes.arq.pessoa,1,16
  ),id.set.pesquisa,NA)
}

################
### Execução ###
################
if(is.na(id.set.pesquisa)){
 print(paste("O setor ",i,id.set.pessoa," não foi encontrado na pasta de
 entrada."))
 sink("F:/PA/DEDUPP/RELATORIOS/Relatorio_SETORES.txt",append=TRUE)
   print(paste("O arquivo ",i,id.set.pessoa,"não foi encontrado na pasta de
    entrada."))
 sink()
}else{
 print(id.set.pesquisa)
```

```
### leitura ###
assign("pessoa",read.table(paste(id.set.pesquisa,"txt",sep="."),header=TRU
 E,colClasses="character",sep=";",na.strings=""))

if (nrow(pessoa)==0){
 print(paste("Setor ",id.set.pesquisa," tem zero linhas."))
 sink("F:/PA/DEDUPP/RELATORIOS/Relatorio_SETORES.txt",append=TRUE)
 print(paste("Setor ",id.set.pesquisa," tem zero linhas."))
 sink()}
 else{
 ### deduplicação de pessoas ###
 dados<-pessoa
 var.excl<-which(!names(pessoa)%in%var.nomes.pes)
 jaro.num.pes<-which(names(pessoa)%in%jaro.num.pes1)
 var.bloco<-which(names(dados)=="V9001")   ### UV
 rpairs<-compare.dedup(dados,blockfld=var.bloco,exclude=var.excl,
 strcmp=jaro.num.pes,strcmpfun=jarowinkler)
 comp.dados<-rpairs$pairs
 idade1<-as.numeric(dados[comp.dados[,"id1"],"V6033"])
 idade2<-as.numeric(dados[comp.dados[,"id2"],"V6033"])
 escores.idade<-peso.idade(idade1,idade2)
 escores.idade1<-rep(0,length(escores.idade))
 escores.idade1[escores.idade>=0.1]<-1
 comp.dados<-subset(comp.dados,select=-is_match)
 comp.dados<-as.matrix(comp.dados)
 ind.na<-is.na(comp.dados)
 comp.dados[ind.na]<-0
 comp.dados1<-comp.dados[,-c(1,2)]
 for(j in 1:ncol(comp.dados1)){
  comp.dados1[,j]<-(comp.dados1[,j]>=patamar.jaro.pes[j])*1
 }
 comp.dados1[,"V6033"]<-escores.idade1
 comp.dados[,-c(1,2)]<-comp.dados1
 mesma_det<-apply(comp.dados1,1,mesma.pess.dedup)
 comp.dados<-cbind(comp.dados,mesma_det)
 comp.dados.CE<-as.data.frame(comp.dados)
 comp.dados.CE<-subset(comp.dados.CE,mesma_det==1)
 if(!nrow(comp.dados.CE)>0){
 print(paste("O Setor ",id.set.pesquisa," não possui pessoa duplicada"))
 sink("F:/PA/DEDUPP/RELATORIOS/Relatorio_SETORES.txt",append=TRUE)
 print(paste("O Setor ",id.set.pesquisa," não possui pessoa duplicada"))
 sink()
 }else{
  paresduplosCE<-cbind(1,dados[comp.dados.CE$id1,],dados[comp.dados.CE$
   id2,])
  paresduplosCE<-duplica.linhas(paresduplosCE,c("UM",names(dados)))
  duplas<-subset(paresduplosCE,select=-UM)

  ### saída em .xls
  nomes<-list.files("F:/PA/DEDUPP/DEDUPLICADOS")
 if(paste("DedupCompleto",i,id.set.pessoa,".xls",sep="")%in%nomes){
 file.remove(paste("F:/PA/DEDUPP/DEDUPLICADOS/DedupCompleto",i,id.set.pess
  oa,".xls",sep=""))}
 channel2 <- odbcConnectExcel(paste("F:/PA/DEDUPP/DEDUPLICADOS/DedupCompl
  eto",i,id.set.pessoa,".xls",sep=""),readOnly = FALSE)
 sqlSave(channel2,duplas,"duplas",rownames=FALSE)
 odbcCloseAll()

 ### saída em .txt para atualização do banco de dados
 duplas2<-subset(duplas,select=V9011)
 write.table(duplas2,file=paste("F:/PA/DEDUPP/DEDUPLICADOS/Dedup_",i,id.
  set.pessoa,".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE)
 }
```

```
 } # Fim deduplicação

 ### Copiar arquivos para a pasta de PROCESSADOS e remover da pasta de
 ### origem
  file.copy(from=paste("F:/PA/DEDUPP/",i,id.set.pessoa,".txt",sep=""),
   to=paste("F:/PA/DEDUPP/PROCESSADOS/",i,id.set.pessoa,".txt",sep=""),over
write=TRUE)
  file.remove(paste("F:/PA/DEDUPP/",i,id.set.pessoa,".txt",sep=""))
 }
}
```

## Anexo 5 – Pareamento de domicílios usando setores vizinhos

```
#############################################################################
###                            IBGE/DPE/COMEQ                             ###
###        Pareamento de domicílios e pessoas do CENSO e da PA            ###
###                            Setembro/2010                              ###
#############################################################################
###             PAREAMENTO PROBABILÍSTICO (EM) - DOMICÍLIOS               ###
#############################################################################
### Autores: Djalma Pessoa                                                ###
###          Fábio Figueiredo                                             ###
###          Vinicius Xavier                                              ###
#############################################################################

tempo<-Sys.time()

###############
### Funções ###
###############
library(RecordLinkage)
library(clue)
library(sendmailR)
library(RODBC)
source("F:/PA/PROGRAMAS R/PesqAval.r")

#id.set<-commandArgs(TRUE)

## leitura do arquivo que contém todos os vizinhos
setwd("F:/PA/")
assign("Teste",read.table("resultado_basico_vizinho sem set vazio.csv",
 header=TRUE,colClasses="character",sep=";",na.strings=""))
## leitura do arquivo que contém todos os pares já formados
assign("sem.par",read.table("uv_pa_sem_par.csv",header=TRUE,colClasses=
 "character",sep=",",na.strings=""))
Fvizinho<-function(id.set){
pos<-sem.par[,1] %in% id.set
sem.par.id.set<-sem.par[pos==TRUE,]
V9011sempar<-paste(sem.par.id.set[,1],sem.par.id.set[,2],sep="")
print(id.set)

#identificação dos vizinhos
pos<-Teste[,1]%in%id.set
vizinho<-Teste[pos==TRUE,]

#identificação de todos os arquivos da pasta
nomes.arq<-list.files("F:/PA/VIZINHO/")
nomes.arq<-nomes.arq[grep("txt",nomes.arq)]
setwd("F:/PA/VIZINHO/")

## leitura dos arquivos vizinhos
dados<-c()
for(i in vizinho[,2]){
i<-paste("C",i,".txt",sep="")
assign("vizTemporario",read.table(i,header=TRUE,colClasses="character",
 sep=";",na.strings=""))
dados<-rbind(dados,vizTemporario)
}


#################
### Variáveis ###
#################
```

```
var.nomes<-c("TIPO","TIT_NOME","NUMERO_VAL","COMPLEMENTO","V9004","V9006",
 "V9008","V9009")
jaro.num<-which(var.nomes%in%c("TIT_NOME","V9004","V9006"))
patamar.jaro<-c(1,0.90,1,1,0.90,0.90,1,1)

#sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",
#         id.set,".txt",sep=""),append=TRUE)
# cat("\n",substring(tempo,1,19),"\n",sep="")
# cat("Setor:","\n",sep="")
# sink()
#
tempo<-gsub(":","_",tempo)

##########################
### Preâmbulo          ###
##########################

  list.CE<-paste("C",id.set,sep="")
  list.PA<-paste("P",id.set,sep="")

  ########################
  ### Execução         ###
  ########################
  resumo<-matrix(NA,length(id.set),6)
  dimnames(resumo)<-list(id.set,c("n_CE","n_PA","n_Pares","%_Pares",
   "n_RES_CE","n_RES_PA"))
  list.res<-list()
  list.id<-list()
  nreg<-matrix(NA,nrow=length(id.set),ncol=2)
  dados.CE0<-dados
  setwd("F:/PA/VIZINHO/TODOS SETORES/")
  assign("dados.PA0",read.table(paste(list.PA,"txt",sep="."),
   header=TRUE,colClasses="character",sep=";",na.strings=""))
  if(nrow(dados.CE0)>0 & nrow(dados.PA0)>0){
   dados.CE0[is.na(dados.CE0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
   dados.PA0[is.na(dados.PA0[,"PTO_REF"]),"PTO_REF"]<-"S/REF"
   dados.CE0[,"V4001"]<-as.integer(dados.CE0[,"V4001"])
   dados.PA0[,"V4001"]<-as.integer(dados.PA0[,"V4001"])
   dados.CE0<-subset(dados.CE0,V4001==1)
   dados.PA0<-subset(dados.PA0,V4001==1)
   dados1<-dados.CE0[,var.nomes]
   dados2<-dados.PA0[,var.nomes]
   nreg[1,]<-c(nrow(dados1),nrow(dados2))
   ## uso do nome do cônjuge
   nom.conjCE<-dados.CE0[,"V9007"]
   nom.conjPA<-dados.PA0[,"V9007"]
   CE.V9007.P<-unlist(lapply(strsplit(nom.conjCE," "),function(t)t[1]))
   PA.V9007.P<- unlist(lapply(strsplit(nom.conjPA," "),function(t)t[1]))
   CE.V9007.U<-unlist(lapply(strsplit(nom.conjCE," "),function(t)t[length(
    t)]))
   PA.V9007.U<-unlist(lapply(strsplit(nom.conjPA," "),function(t)t[length(
    t)]))
   CE.VAUX1<-paste(dados1$V9004,CE.V9007.P,sep="")
   CE.VAUX2<-paste(dados1$V9004,CE.V9007.P,sep="")
   PA.VAUX1<-paste(dados2$V9004,PA.V9007.P,sep="")
   PA.VAUX2<-paste(PA.V9007.P,dados2$V9004,sep="")
   AUX.CE<-data.frame(VAUX1= CE.VAUX1, VAUX2= CE.VAUX2)
   AUX.PA<-data.frame(VAUX1= PA.VAUX1, VAUX2= PA.VAUX2)
   pares.aux<-compare.linkage(AUX.CE,AUX.PA,strcmp=c(1,2),strcmpfun=jarowin
    kler)$pairs
   conj.jaro<-pmax(pares.aux$VAUX1,pares.aux$VAUX2)
```

```
    ### Excluir variáveis ###
    var.num<-varexcl(dados1, dados2,jaro.num,patamar.jaro)
    var.nomes1<-var.nomes[var.num]
    jaro.num<-which(var.nomes1%in%c("TIT_NOME","V9004","V9006"))
    patamar.jaro=patamar.jaro[var.num]
    dados1<-dados.CE0[,var.nomes1]
    dados2<-dados.PA0[,var.nomes1]

    ### Relatório em txt ###
    dados.CE<-dados.CE0[,var.nomes]
    dados.PA<-dados.PA0[,var.nomes]
    dif.nom.CE<-lapply(dados.CE,unique)
    tab.nom.CE<-lapply(dados.CE,table)
    dif.nom.PA<-lapply(dados.PA,unique)
    tab.nom.PA<-lapply(dados.PA,table)
#    sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",id.set,".txt",sep=""),
#     append=TRUE)
#    cat("\nRelatório do Setor:",id.set,"\n")
#    cat("\nNomes distintos da variável no Censo:","\n")
#    print(tab.nom.CE)
#    cat("\nNomes distintos da variável na PA:","\n")
#    print(tab.nom.PA)
#    cat("\nVariáveis utilizadas:","\n")
#    print(var.nomes1)
#    sink()
#
    ### Pareamento ###
    res.pares<-pareia.em(dados1=dados1,dados2=dados2,urb=urb,strcomp=
     jaro.num,patamar=patamar.jaro,criterio=4,conj.aux=conj.jaro)
    Pares.em<-res.pares[[1]]

    ### Se fez pares -> Redução 1 a 1, senão -> Relatório ###
    if(nrow(Pares.em)>0){
      ID_CE<-Pares.em[,"id1"]
      ID_PA<-Pares.em[,"id2"]
      var.com<-intersect(names(dados.PA0),names(dados.CE0))
      pares.finais<-cbind(Escore=Pares.em[,"Escore"],Conc=Pares.em[,"Conc"],
       ID_CE,dados.CE0[ID_CE,var.com],Conc=Pares.em[,"Conc"],ID_PA,dados.PA0
       [ID_PA,var.com])
      pares.finais<-pares.finais[order(pares.finais[,"Escore"],decreasing=
       TRUE),]

    ## identificação das UV que não tinham par
      paresnovos<-pares.finais[,30]%in%V9011sempar
      pares.finais<-pares.finais[paresnovos,]
      resumo[1,1]<-nrow(dados1)
      resumo[1,2]<-length(V9011sempar)

    if(nrow(pares.finais)>0){
      list.id[[1]]<-paste(pares.finais[,5],pares.finais[,30],sep=".")
      duplas<-duplica.linhas(pares.finais,c("Escore","Conc","ID",var.com))
      list.res[[1]]<-list(duplas,res.pares[[2]],res.pares[[3]])

      ### Obtenção dos resíduos ###
      res.PA<-setdiff(1:nrow(dados.PA0),pares.finais[,"ID_PA"])
      res.CE<-setdiff(1:nrow(dados.CE0),pares.finais[,"ID_CE"])
      resA<-cbind(ID_CE=res.CE,dados.CE0[res.CE,])
      resB<-cbind(ID_PA=res.PA,dados.PA0[res.PA,])

      ### Exportando os pares no formato xls ###

      duplas<-duplas[,-c(1,2,3,4,6,7,9,10,12,13,14,17,19,20,21,23,24,25,26)]
```

```
    channel2 <-
odbcConnectExcel(paste("F:/PA/VIZINHO/PAREADOS/",id.set,".xls",sep=""),read
Only = FALSE)
   sqlSave(channel2,duplas,"duplas",rownames=FALSE)
   odbcCloseAll()

   duplas<-subset(duplas,select=V9001)
     ### Exportando os pares no formato txt ###
     write.table(duplas,file=paste("F:/PA/VIZINHO/PAREADOS/",
     id.set,".txt",sep=""),sep=";",dec=",",quote=FALSE,row.names=FALSE,col.
names=FALSE)

     ### Resumo do pareamento ###

     resumo[1,3]<-nrow(pares.finais)
     resumo[1,4]<-round(100*resumo[1,3]/min(resumo[1,1],resumo[1,2]),1)
     resumo[1,5]<-nrow(resA)
     resumo[1,6]<-resumo[1,2]- resumo[1,3]
   }}else{
     resumo[1,1]<-nrow(dados1)
     resumo[1,2]<-nrow(dados2)
     resumo[1,3]<-0
     resumo[1,4]<-0
     resumo[1,5]<-nrow(dados.CE0)
     resumo[1,6]<-nrow(dados.PA0)
     resA<-dados.CE0
     resB<-dados.PA0
   }

   ### Relatório em formato txt ###
   intercPACE<-integer(length(var.nomes))
   mat.crit<-matr.crit<-matrix(NA,3,length(var.nomes))
   dimnames(mat.crit)<-list(c("dif_CE","dif_PA","dif_int"),var.nomes)
   for(j in 1:length(var.nomes))intercPACE[j]<-length(intersect(dif.nom.CE[
    [j]],dif.nom.PA[[j]]))
   mat.crit[1,]<-sapply(dif.nom.CE,length)
   mat.crit[2,]<-sapply(dif.nom.PA,length)
   mat.crit[3,]<-intercPACE

#  sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",id.set,".txt",sep=""),
#   append=TRUE)
#   cat("\nFrequências das interseções para Censo e PA:","\n")
#   print(mat.crit)
#   cat("\nResumo:","\n")
#   print(resumo[1,])
#   sink()
#    sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",id.set,".txt",sep=""),
#     append=TRUE)
#   cat(id.set,".....................ok","\n",sep="")
#   sink()
  }else{
#  sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",
#         id.set,".txt",sep=""),append=TRUE)
#   cat(id.set,".........Não processado","\n",sep="")
#   sink()
  }

 # resumo<-cbind(resumo,AVISO=aprov(resumo))

 ### Copiar arquivos para a pasta de PROCESSADOS e remover da pasta de
 ### origem
 file.copy(from=paste("F:/PA/VIZINHO/TODOS SETORES/","P",id.set,".txt",
  sep=""),to=paste("F:/PA/VIZINHO/PROCESSADOS/","P",id.set,".txt",sep=""),
```

```
   overwrite=TRUE)
  file.remove(paste("F:/PA/VIZINHO/TODOS SETORES/","P",id.set,".txt",
   sep=""))
  if(nrow(pares.finais)>0){
  msg<-paste(" SETOR    ",dimnames(resumo)[[2]][1],dimnames(resumo)[[2]]
   [2],dimnames(resumo)[[2]][3],dimnames(resumo)[[2]][4],dimnames(resumo)
   [[2]][5],dimnames(resumo)[[2]][6],"  ",sep=" ")

    # sink(paste("F:/PA/VIZINHO/RELATORIOS/relatorio",id.set,".txt",
    # sep=""),append=TRUE)
  ### Gerar arquivo de resumo do lote ###
  write.table(resumo,paste("F:/PA/VIZINHO/RELATORIOS/resumo",id.set,".txt",
   sep=""),quote=FALSE)
  ### Enviar e-mail com o resumo do lote ###
  from<-"R"
  to<-"PA2010@ibge.gov.br"
  subject<-paste("Vizinhos do: " ,row.names(resumo)[1],sep="")
  for(i in 1:nrow(resumo)){
   msg<-paste(msg,"\n",formata(set=row.names(resumo)[i], x=resumo[i,]))
  }
  mail<-sendmail(from,to,subject,msg,control=list(smtpServer="172.31.0.
   107"))
}
}

todos.id.set<-list.files("F:/PA/VIZINHO/TODOS SETORES/")
todos.id.set<-todos.id.set[grep("P",todos.id.set)]
todos.id.set<-substring(todos.id.set,2,16)

for(id.set in todos.id.set){
vizinho<-try(Fvizinho(id.set))
if(!is.logical(vizinho)) {
write.table(id.set,file="F:/PA/VIZINHO/PAREADOS/naoprocessados.txt",
sep=";",dec=",",quote=FALSE,row.names=FALSE,col.names=FALSE,append=TRUE)
}}
```

## Anexo 6 – Funções auxiliares utilizadas

```
##########################################################################
###                          IBGE/DPE/COMEQ                            ###
###         Pareamento de domicílios e pessoas do CENSO e da PA        ###
###                           Setembro/2010                            ###
##########################################################################
###                        BIBLIOTECAS/FUNÇÕES                         ###
##########################################################################
### Autores: Djalma Pessoa                                             ###
###          Fábio Figueiredo                                          ###
###          Vinicius Xavier                                           ###
##########################################################################

#############
# Domicílio #
#############

##############################
### Aviso de erro do setor ###
##############################
aprov<-function(x){
 if(!is.na(x[3])& as.numeric(x[3])/min(as.numeric(x[1]),as.numeric(x[2]))>
  .10)
 "Aprovado"else
 "Reprovado"
}
#########################################################
#                        varexcl                        #
#########################################################
#        Excluir variáveis pouco informativas           #
#########################################################
varexcl<-function(dados1,dados2,strcmp,cutoff){
 varuteis<-1:ncol(dados1)
 names(varuteis)<-colnames(dados1)
 rpairs<-compare.linkage(dados1,dados2,strcmp=strcmp,strcmpfun=jarowinkler)
 pares=rpairs$pairs
 pares=pares[,-c(1,2,ncol(pares))]
 pares=as.matrix(pares)
 pares[is.na(pares)]=0
 for (i in 1:ncol(pares)) pares[,i]<-(pares[,i]>=cutoff[i])*1
 ### Testa se não há concordância nas variáveis (Todas)
 vec.concord=apply(pares,2,function(t)length(unique(t)))
 var.excl=which(vec.concord==1)
 if(length(var.excl)>0)varuteis<-(varuteis)[-var.excl]
 dados1<-dados1[,varuteis]
 dados2<-dados2[,varuteis]
 ## Entropia para as variáveis (todas)
 EntrCE<-lapply(dados1,entropia)
 EntrPA<-lapply(dados2,entropia)
 ## Número de valores distintos (todas)
 TabCE<-lapply(dados1,unique)
 TabPA<-lapply(dados2,unique)
 TabCEComp<-sapply(TabCE,length)
 TabPAComp<-sapply(TabPA,length)
 ##Número de valores comuns de pares das variáveis TIPO e TIT_NOME
 VecCompCom<-integer(length(TabCE))
 names(VecCompCom)<-names(TabCE)
 for(i in 1:length(TabCE))VecCompCom[i]<-length(intersect(TabCE[[i]],
  TabPA[[i]]))
  ncolTIPO<-which(names(TabCEComp)=="TIPO")
```

```
  if("TIPO"%in%names(varuteis)&&(VecCompCom["TIPO"]/max(TabCEComp["TIPO"],
   TabPAComp["TIPO"])<=.5)) varuteis<-varuteis[-ncolTIPO]
  ncolTITNOME<-which(names(varuteis)=="TIT_NOME")
  if("TIT_NOME"%in%names(varuteis)&&(VecCompCom["TIT_NOME"]/max(TabCEComp["
   TIT_NOME"],TabPAComp["TIT_NOME"])<=.5)) varuteis<-varuteis[-ncolTITNOME]
  ### Inclusão das variáveis COMPLEMENTO e NUMERO_VAL

  if("COMPLEMENTO"%in%names(varuteis)&&"NUMERO_VAL"%in%names(varuteis)){
  if(entropia(dados1$COMPLEMENTO)<0.7){
    pos_COMPLEMENTO<- which(names(varuteis)=="COMPLEMENTO")
    varuteis<-varuteis[-pos_COMPLEMENTO]
  }else{
   if (entropia(dados1$NUMERO_VAL)<0.7) {
     pos_NUMERO_VAL<- which(names(varuteis)=="NUMERO_VAL")
     varuteis<-varuteis[-pos_NUMERO_VAL]
   }
  }
}
varuteis
}

#####################################################################
#                          duplica.linhas                           #
#####################################################################
# Função para formatar arquivo de pares                             #
# colocando os pares em linhas consecutivas (formato 2 em 2)        #
#####################################################################
duplica.linhas<-function(dados,nomes=NULL){
 printfun=function(x){
  c(x[1:((length(x)+1)/2)],c("",x[((length(x)+3)/2):length(x)]))
 }
 duplas<-apply(as.matrix(dados),1,printfun)
 duplas<-as.data.frame(matrix(duplas[T],nrow=ncol(duplas)*2,
 ncol=nrow(duplas)/2,byrow=TRUE))
 colnames(duplas)<-nomes
 duplas
}

#####################################################################
#                             rurconc                               #
#####################################################################
#    Função para definir o critério para os setores rurais          #
#####################################################################
rurconc<-function (t)
{
  l<-length(t)
  nconc <- 0
  if (sum(t)>=3 | (t[(l-1)] == 1 & sum(t) >= 1))nconc<-3
  nconc
}

#####################################################################
#                             deduplic                              #
#####################################################################
#    Função para fazer a deduplicação usando algoritmo EM           #
#####################################################################
deduplic<-function(dados,urb=TRUE,strcomp,patamar){
 nomes.var<-colnames(dados)
 num.var<-length(nomes.var)
 rpairs<-compare.dedup(dados,strcmp=strcomp,strcmpfun=jarowinkler)
 comp.dados<-rpairs$pairs
 comp.dados<-subset(comp.dados,select=-is_match)
 comp.dados<-as.matrix(comp.dados)
```

```
 ind.na<-is.na(comp.dados)
 comp.dados[ind.na]<-0
 for(i in 1:num.var){
  comp.dados[,i+2]<-as.array((comp.dados[,i+2]>=patamar[i])*1)
 }
 if(urb)vec.urb<-rowSums(comp.dados[,-c(1,2)])else{
  vec.urb<-apply(comp.dados[,-c(1,2)],1,rurconc)
 }
 if(sum(vec.urb>=3)>0){
  duplas<-comp.dados[vec.urb>=3,]
  id.duplas<-subset(duplas,select=c(id1,id2))
  id.duplas<-cbind(1,id.duplas)
  id.duplas<-duplica.linhas(dados=id.duplas,nomes=c("um","id"))
  id.duplas<-as.numeric(as.character(id.duplas[,"id"]))
 }else{
  id.duplas<-0
 }
 id.duplas
}

#######################################################################
#                              emWeights1                             #
#######################################################################
#  Modificação da função emWeights da biblioteca RecordLinkage       #
#  para calcular os escores, usando patamares distintos para a       #
#  similaridade entre strings pelo critério de Jaro-Winkler          #
#######################################################################
emWeights1<-function(rpairs,urb=TRUE,cutoff=0.95,...){
 if(!("RecLinkData"%in%class(rpairs)||"RecLinkResult"%in%class(rpairs)))
  stop(sprintf("Wrongclassforrpairs:%s",class(rpairs)))
 if(nrow(rpairs$pairs)==0)
  stop("Norecordpairs!")
 if(!is.numeric(cutoff))
  stop(sprintf("Illegaltypeforcutoff:%s",class(cutoff)))
 if(cutoff<0||cutoff>1)
  stop(sprintf("Illegalvalueforcutoff:%g",cutoff))
 pairs=rpairs$pairs
 pairs=pairs[,-c(1,2,ncol(pairs))]
 pairs=as.matrix(pairs)
 pairs[is.na(pairs)]=0
 is_fuzzy=!all(is.element(pairs,0:1))
 ### Modificação introduzida para usar cutoff vetorial ###
 if(is_fuzzy){
  pairs_fuzzy=pairs
  for(i in 1:ncol(pairs)){
   pairs[,i]<-(pairs[,i]>=cutoff[i])*1
  }
 }
 n_data=nrow(pairs)
 observed_count=countpattern(pairs)
 n_attr=ncol(pairs)
 patterns=bincombinations(n_attr)
 x=c(rep(0,nrow(patterns)),rep(1,nrow(patterns)))
 s=c(1:length(observed_count),1:length(observed_count))
 i=rep(1,nrow(patterns))
 X=cbind(i,x,rbind(patterns,patterns),rbind(patterns,patterns)*x)
 u=rpairs$frequencies
 m=0.97
 prob_M=1/sqrt(n_data)*0.1
 init_M=apply(patterns,1,function(a)prod(a*m+(1-a)*(1-m))*n_data*prob_M)
 init_U=apply(patterns,1,function(a)prod(a*u+(1-a)*(1-u))*n_data*(1-
  prob_M))
 expected_count=c(init_U,init_M)
```

```
 res=mygllm(observed_count,s,X,E=expected_count,...)
 n_patterns=length(res)/2
 matchrate=res[(n_patterns+1):(2*n_patterns)]/res[1:n_patterns]
 n_matches=sum(res[(n_patterns+1):(2*n_patterns)])
 n_nonmatches=sum(res[1:n_patterns])
 U=res[1:n_patterns]/n_nonmatches
 M=res[(n_patterns+1):(2*n_patterns)]/n_matches
 W=log(M/U,base=2)
 indices=colSums(t(pairs)*(2^(n_attr:1-1)))+1
 ret=rpairs
 ret$conc=pairs
 ret$counts=observed_count
 ret$M=M
 ret$U=U
 ret$W=W
 ret$Wdata=W[indices]
 if(is_fuzzy){
  str_weights=apply(pairs_fuzzy^pairs,1,prod)
  ret$Wdata=ret$Wdata+log(str_weights,base=2)
 }
 cat("\n")
 return(ret)
}

#####################################################################
#                              pareia.em                            #
#####################################################################
#  Função para executar o pareamento usando algoritmo EM utilizando #
#  diferentes                                                       #
#  critérios e redução 1:1                                          #
#####################################################################
pareia.em<-function(dados1,dados2,urb=TRUE,strcomp,patamar,criterio=2,
 conj.aux){
 if(!criterio%in%c(1:4))stop("CRITÉRIOINVÁLIDO!")

 #criterio=2
 nA<-nrow(dados1)
 nB<-nrow(dados2)
 nmaxpar<-min(nA,nB)
 rpairsPACE<-compare.linkage(dados1,dados2,strcmp=strcomp,strcmpfun=
  jarowinkler)
 rpairsPACE$pairs[,"V9004"]<-conj.aux
 emWeightsPACE<-emWeights1(rpairsPACE,urb=urb,cutoff=patamar)
 tab.esco<-table(emWeightsPACE$Wdata)

 ### associação 1:1 ###
 emWeightsPACE$Wdata[emWeightsPACE$Wdata==Inf]<-max(emWeightsPACE$Wdata[
  emWeightsPACE$Wdata!=Inf])
 emWeightsPACE$Wdata[emWeightsPACE$Wdata==-Inf]<-min(emWeightsPACE$Wdata[
  emWeightsPACE$Wdata!=-Inf])
 w<-emWeightsPACE$Wdata-min(emWeightsPACE$Wdata)
 mat_escores<-matrix(0,nA,nB,byrow=TRUE)
 mat_escores1<-matrix(0,nA,nB,byrow=TRUE)
 mat_conc<-matrix(0,nA,nB,byrow=TRUE)
 mat_conc1<-matrix(0,nA,nB,byrow=TRUE)
 mat.urb<-emWeightsPACE$conc
 mat.urb.id<-cbind(id1=emWeightsPACE$pairs$id1,id2=emWeightsPACE$pairs$id2,
  emWeightsPACE$conc)
 vec.urb<-rowSums(mat.urb)
 vec.urb1<-rowSums(mat.urb)
 vec.urb.id<-cbind(vec.urb,vec.urb1,id1=emWeightsPACE$pairs[,"id1"],id2=
  emWeightsPACE$pairs[,"id2"],w,Wdata=emWeightsPACE$Wdata)
 colnames(vec.urb.id)[3:4]<-c("id1","id2")
```

```
if(criterio==1){
 nmaxpar<-min(nA,nB)
 tab.esco<-table(emWeightsPACE$Wdata)
 nsup<-head(rev(which(cumsum(rev(tab.esco))<=nmaxpar)),1)
 pto.corte1<-as.numeric(names(nsup))
 pto.corte2<-as.numeric(names(rev(tab.esco)[nsup+1]))
 ncorte1<-sum(emWeightsPACE$Wdata>=pto.corte1)
 ncorte2<-sum(emWeightsPACE$Wdata>=pto.corte2)
 pto.corte<-pto.corte1
 if(abs(ncorte1-nmaxpar)>abs(ncorte2-nmaxpar))pto.corte<-pto.corte2
 paresPACE<-emClassify(emWeightsPACE,threshold.upper=pto.corte)
 indM<-which(paresPACE$prediction=="L")
 vec.urb.id<-vec.urb.id[indM,]
}
### Critério 3 - Ponto de corte pelo número de concordâncias ###
if(criterio==3)vec.urb.id<-vec.urb.id[vec.urb>=3,]
mat.ind<-vec.urb.id[,c("id1","id2")]
mat_escores[mat.ind]<-vec.urb.id[,"w"]
mat_escores1[mat.ind]<-vec.urb.id[,"Wdata"]
mat_conc[mat.ind]<-vec.urb.id[,"vec.urb"]
mat_conc1[mat.ind]<-vec.urb.id[,"vec.urb1"]
if(nA>nB){
 mat_0<-matrix(0,nA,nA-nB)
 mat_escores<-cbind(mat_escores,mat_0)
}else{
 mat_0<-matrix(0,nB-nA,nB)
 mat_escores<-rbind(mat_escores,mat_0)
}
result<-solve_LSAP(mat_escores,maximum=TRUE)
data.id.final<-cbind(seq_along(result),result)
dimnames(data.id.final)<-list(NULL,c("id1","id2"))
if(nA>nB)data.id.final<-data.id.final[data.id.final[,"id2"]<=nB,]else{
 data.id.final<-data.id.final[data.id.final[,"id1"]<=nA,]
}
ind.final<-mat_escores[data.id.final]>0
data.id.final<-data.id.final[ind.final,]
Pares<-data.frame(Escore=mat_escores1[data.id.final],Conc1=mat_conc[
 data.id.final],
Conc=mat_conc1[data.id.final],id1=data.id.final[,"id1"],dados1[
 data.id.final[,"id1"],],
id2=data.id.final[,"id2"],dados2[data.id.final[,"id2"],])
colnames(Pares)<-c("Escore","Conc1","Conc","id1",paste(colnames(dados1),
 ".1",sep=""),"id2",paste(colnames(dados2),".2",sep=""))
ind.Pares<-cbind(Pares[,"id1"],Pares[,"id2"])
colnames(ind.Pares)<-c("id1","id2")
mat.conc.pares<-merge(ind.Pares,mat.urb.id,all.x=TRUE)
mat.conc.pares1<-mat.conc.pares[order(mat.conc.pares$id1),]
if(!"V9004"%in%colnames(mat.conc.pares1))mat.conc.pares1[,"V9004"]<-0
if(!"V9006"%in%colnames(mat.conc.pares1))mat.conc.pares1[,"V9006"]<-0
if(!"NUMERO_VAL"%in%colnames(mat.conc.pares1))mat.conc.pares1[,
 "NUMERO_VAL"]<-0
if(!"V9008"%in%colnames(mat.conc.pares1))mat.conc.pares1[,"V9008"]<-0
if(!"V9009"%in%colnames(mat.conc.pares1))mat.conc.pares1[,"V9009"]<-0
if(!"COMPLEMENTO"%in%colnames(mat.conc.pares1))mat.conc.pares1[,
 "COMPLEMENTO"]<-0
if(criterio==4) Pares<-subset(Pares,Conc1>6|(Conc1==5&mat.conc.pares1[,
 "V9004"]==1)|(mat.conc.pares1[,"V9004"]==1&Conc1>=4&mat.conc.pares1[,
 "NUMERO_VAL"]==1)|mat.conc.pares1[,"V9004"]==1&Conc1>=4&mat.conc.pares1[,
 "V9006"]==1))
if(criterio==1) Pares<-subset(Pares,Conc1>=5|(mat.conc.pares1[,"V9004"]==
 1&Conc1==4)|( mat.conc.pares1[,"V9004"]==1&mat.conc.pares1[,"V9006"]==
 1&mat.conc.pares1[,"NUMERO_VAL"]==1)|mat.conc.pares1[,"V9004"]==1&mat.
```

```
  conc.pares1[,"V9006"]==1&mat.conc.pares1[,"V9008"]==1&mat.conc.Pares1[,
  "V9009"]==1)|(mat.conc.pares1[,"V9004"]==1&mat.conc.pares1[,"V9006"]==1&
  mat.conc.pares1[,"COMPLEMENTO"]==1))
 if(criterio==2){
 if(!"V9006"%in%names(mat.conc.pares1)){
  Pares<-subset( Pares,(Conc1>=3)|(mat.conc.pares1[,"V9004"]==1))
 }else{
  Pares<-subset( Pares,(Conc1>=3)|(mat.conc.pares1[,"V9004"]==1)|(Conc1>=
   2&mat.conc.pares1[,"V9006"]==1))}
 }
  Pares<-subset(Pares,select=-Conc1)

 list(Pares=Pares,contagens=emWeightsPACE$counts,Escores=unique(sort(
  emWeightsPACE$Wdata)))
}

############
# Pessoas #
############

#########################################################################
#                               peso.idade                              #
#########################################################################
#     Função para comparar idades (escore entre 0 e 1)                  #
#########################################################################
peso.idade<-function(valor1,valor2,dif.max=5){
 valor1[valor1>200]<-0
 valor2[valor2>200]<-0
 peso.conc.parc=1-abs(valor1-valor2)/(dif.max+1)
 peso.conc.parc[peso.conc.parc<0]<-0
 peso.conc.parc
}

#########################################################################
#                                esc.pess                               #
#########################################################################
#  Função para atribuir escore para as configurações definidas          #
#########################################################################
esc.pess<-function(t){
 escore<-0
 if(sum(t)==length(t))escore<-1
 if(t[1]==1&t[2]==0&t[3]==1)escore<-0.8
 if(t[1]==1&t[2]==1&t[3]==0)escore<-0.7
 escore
}

#########################################################################
#                               n.max.pares                             #
#########################################################################
#   Função para determinar o número máximo de pares possíveis           #
#########################################################################
n.max.pares<-function(dados){
 rotA<-unique(dados[,"id1"])
 rotB<-unique(dados[,"id2"])
 nA<-length(rotA)
 nB<-length(rotB)
 n<-min(nA,nB)
 n
}

#########################################################################
#                            mesma.pess.dedup                           #
#########################################################################
```

```
#  Função para determinar os critérios na deduplicação de pessoas   #
#####################################################################
mesma.pess.dedup<-function(t){
 ind<-0
 if(sum(t)==(length(t)-1)&t[1]==1) ind<-1
 if(sum(t)==length(t)) ind<-1
 ind
}

#####################################################################
#               assoc1:1                                            #
#####################################################################
#         Função para fazer associação 1 a 1                        #
#####################################################################
assoc11<-function(dados){
 if(sum(dados[,"escores_det"])>0){
  rotA<-unique(dados[,"id1"])
  rotB<-unique(dados[,"id2"])
  nA<-length(rotA)
  nB<-length(rotB)
  if(min(nA,nB)==1){
   dados<-dados[order(dados[,"escores_det"],decreasing=TRUE),]
   mat.id.final<-unlist(dados[1,c("id1","id2"),drop=TRUE])
   mat.id.final<-matrix(mat.id.final,nrow=1)
  }else{
   mat.escores<-matrix(0,nA,nB)
   dimnames(mat.escores)<-list(as.character(rotA),as.character(rotB))
   for(i in 1:nrow(dados))mat.escores[as.character(dados[i,"id1"]),
   as.character(dados[i,"id2"])]<-as.numeric(dados[i,"escores_det"])
   ### tornando a matriz de escores quadrada ###
   if(nA>nB){
    mat0<-matrix(0,nA,nA-nB)
    mat.escores<-cbind(mat.escores,mat0)
   }else{
    mat0<-matrix(0,nB-nA,nB)
    mat.escores<-rbind(mat.escores,mat0)
   }
   result<-solve_LSAP(mat.escores,maximum=TRUE)
   data.id.final<-cbind(seq_along(result),result)
   dimnames(data.id.final)<-list(NULL,c("id1","id2"))
   if(nA>nB)data.id.final<-data.id.final[data.id.final[,"id2"]<=nB,]else{
    data.id.final<-data.id.final[data.id.final[,"id1"]<=nA,]
   }
   ind.final<-mat.escores[data.id.final]>0
   data.id.final<-data.id.final[ind.final,,drop=FALSE]
   id1<-dimnames(mat.escores)[[1]][data.id.final[,1]]
   id1<-as.integer(id1)
   id2<-dimnames(mat.escores)[[2]][data.id.final[,2]]
   id2<-as.integer(id2)
   mat.id.final<-matrix(NA,nrow(data.id.final),2)
   for(i in 1:nrow(data.id.final)){
    mat.id.final[i,]<-c(id1[i],id2[i])
   }
  }
 }else{
  mat.id.final<-matrix(c(-1,-1),nrow=1)
 }
 dimnames(mat.id.final)<-list(NULL,c("id1", "id2"))
 mat.id.final
}
####################
# Formata mensagem #
####################
```

```
formata<-function(set,x)
 {
   branco<-"     "
   n = length(as.vector(x))
   retorna = character(n)
   a = " "
   for(i in 1:(n-1) )
   {
    if(is.na(x[i]))
    {
      a = ""
      d = 3
      x[i] = "---"
    }else{
        d = nchar(x[i])
    }
    retorna[i] = paste(substr(branco,1,5-d),x[i],sep="")
   }
   retorna[n] = paste(a,x[n],sep="")
   result = paste(set," ", retorna[1], retorna[2],"    ", retorna[3],"  ",
    retorna[4],"  ", retorna[5],"  ", retorna[6],"   ", retorna[7],"   ",
    retorna[8],sep="")
   result
 }

 ###################################################################
 # Exclusão de variáveis com número de valores muito diferentes #
 ###################################################################
  excl.num.val<-function(dados1,dados2){
   dif.nom.1<-lapply(dados1,unique)
   dif.nom.2<-lapply(dados2,unique)
   mat.dist<-matrix(NA,2,ncol(dados1))
   mat.dist[1,]<-sapply(dif.nom.1,length)
   mat.dist[2,]<-sapply(dif.nom.2,length)
   mat.prob<-apply(mat.dist,2,function(t)t/sum(t))
   dist.vec<-apply(mat.prob,2,function(t)max(abs(t-1/length(t))))
   var.num.excl<-which(dist.vec==max(dist.vec))(1:ncol(dados1))[-var.num.
    excl]
  }

#########################################################
#                        entropia                       #
#########################################################
#                 Cálculo da entropia                   #
#########################################################
  entropia<-function(variav){
   variav[is.na(variav)]<-0
   K<-length(table(variav))
   relFreq<-table(variav)/sum(table(variav))
   if(length(relFreq)==1)ent<-0 else
   ent<--sum(relFreq*log(relFreq,K),na.rm = TRUE)
   ent
  }
```

## Textos para Discussão já publicados

### Antiga série

- Pesquisas Contínuas da Indústria - Vol. 1, **nº 1**, janeiro 1988
- Pesquisas Agropecuárias Contínuas: Metodologia - Vol. I, **nº 2**, 1988
- Uma Filosofia de Trabalho: As experiências com o SNIPC e com o SINAPI - Vol. I, **nº 3**, março 1988
- O Sigilo das Informações Estatísticas: Idéias para reflexão - Vol. I, **nº 4**, abril 1988
- Projeções da População Residente e do Número de Domicílios Particulares Ocupados: 1985-2020 - Vol. I, **nº 5**, maio 1988
- Classificação de Atividades e Produtos, Matérias-Primas e Serviços Industriais: Indústria Extrativa Mineral e de Transformação - Vol. 1, **nº 6**, agosto 1988
- A Mortalidade Infantil no Brasil nos Anos 80 - Vol. I, **nº 7**, setembro 1988
- Principais Características das Pesquisas Econômicas, Sociais e Demográficas - Vol. I, **número especial**, outubro 1988
- Ensaio sobre o Produto Real da Agropecuária - Vol. I, **nº 9**, setembro 1988
- Novo Sistema de Contas Nacionais, Ano Base 1980 - Resultados Provisórios - Vol. I**, nº 10**, dezembro 1988
- Pesquisa de Orçamentos Familiares - Metodologia para Obtenção das Informações de Campo - **nº 11**, janeiro 1989
- De Camponesa a Bóia-fria: Transformações do trabalho feminino - **nº 12**, fevereiro 1989
- Pesquisas Especiais do Departamento de Agropecuária - Metodologia e Resultados **- nº 13**, fevereiro 1989
- Brasil - Matriz de Insumo-Produto - 1980 - **nº 14**, maio 1989
- As Informações sobre Fecundidade, Mortalidade e Anticoncepção nas PNADs - **nº 15**, maio 1989
- As Estatísticas Agropecuárias e a III Conferência Nacional de Estatística - **nº 16**, junho 1989
- Brasil - Sistema de Contas Nacionais Consolidadas - **nº 17**, agosto 1989
- Brasil - Produto Interno Bruto Real Trimestral - Metodologia - **nº 18**, agosto 1989
- Estatísticas e Indicadores Sociais para a Década de 90 - **nº 19**, setembro 1989
- Uma Análise do Cotidiano da Pesquisa no DEREN (As Estatísticas do Trabalho) - **nº 20**, outubro 1989
- Coordenação Estatística Nacional - Reflexões sobre o caso Brasileiro - **nº 21**, novembro 1989
- Pesquisa Industrial Anual 1982/84 - Análise dos Resultados - **nº 22**, novembro 1989
- O Departamento de Comércio e Serviços e a III Conferência Nacional de Estatística **- nº 23**, dezembro 1989
- Um projeto de Integração para as Estatísticas Industriais - **nº 24**, dezembro 1989
- Cadastro de Informantes de Pesquisas Econômicas - **nº 25**, janeiro 1990
- Ensaios sobre a Produção de Estatística - **nº 26**, janeiro 1990
- O Espaço das Pequenas Unidades Produtivas: Uma tentativa de delimitação - **nº 27**, fevereiro 1990
- Uma Nova Metodologia para Correção Automática no Censo Demográfico Brasileiro: Experimentação e primeiros resultados - **nº 28**, fevereiro 1990
- Notas Técnicas sobre o Planejamento de Testes e Pesquisas Experimentais - **nº 29**, março 1990

- Estatísticas, Estudos e Análises Demográficas - Uma visão do Departamento de População - **nº 30**, abril 1990
- Crítica de Equações de Fechamento de Empresas no Censo Econômico de 1985 - **nº 31**, maio 1990
- Efeito de Conglomeração da Malha Setorial do Censo Demográfico de 1980 - **nº 32**, maio 1990
- A Redução da Amostra e a Utilização de Duas Frações Amostrais no Censo Demográfico de 1990 - **nº 33**, junho 1990
- Estudos e Pesquisas de Avaliação de Censos Demográficos - 1970 a 1990 - **nº 34**, julho 1990
- A Influência da Migração no Mercado de Trabalho das Capitais do Centro-Oeste - 1980 - **nº 35**, agosto 1990
- Pesquisas de Conjuntura: Discussão sobre Variáveis a Investigar - **nº 36**, setembro 1990
- Um Modelo para Estimar o Nível e o Padrão da Fecundidade por Idade com Base em Parturições Observadas - **nº 37**, outubro 1990
- A Estrutura Operacional de Uma Pesquisa por Amostra - **nº 38**, novembro 1990
- Produção Agrícola, Agroindustrial e de Máquinas e Insumos Agrícolas no Anos 80: Novas Evidências Estatísticas - **nº 39**, dezembro 1990
- A Inflação Medida pelo Índice de Precos ao Consumidor - **nº 40**, janeiro 1991
- A Participação Política Eleitoral no Brasil - 1988, Análise Preliminar - **nº 41**, fevereiro 1991
- Ensaios sobre Estatísticas do Setor Produtivo - **nº 42**, março 1991
- A Produção Integrada de Estatística e as Contas Nacionais: Agenda para Formulação de um Novo Plano Geral de Informações Estatísticas e Geográficas - **nº 43**, março 1991
- Matriz de Fluxos Migratórios Intermunicipais - Brasil - 1980 - **nº 44**, abril 1991
- Fluxos Migratórios Intrametropolitanos - Brasil - 1970-1980 - **nº 45**, abril 1991
- A Revisão da PNAD, A Questão Conceitual e Relatório das Contribuições - **nº 46**, maio 1991
- A Dimensão Ambiental no Sistema de Contas Nacionais - **nº 47**, maio 1991
- Estrutura das Contas Nacionais Brasileiras - **nº 48**, junho 1991
- Mercado do Couro e Resultados da Pesquisa Anual do Couro - **nº 49**, junho 1991
- As Estatísticas e o Meio Ambiente - **nº 50**, julho 1991
- Novo Sistema de Contas Nacionais Séries Correntes: 1981-85 Metodologia, Resultados Provisórios e Avaliação do Projeto - **nº 51**, julho 1991 (2 Volumes: Volume 1 - Metodologia, Resultados Provisórios e Avaliacão do Projeto; Volume 2-Tabelas)
- O Censo Industrial de 1985 - Balanço da Experiência - **nº 52**, agosto 1991
- Análise da Inflação Medida Pelo INPC 1989 - **nº 53**, agosto 1991
- Revisão da PNAD: A Questão Amostral: Módulo II do Anteprojeto - **nº 54**, setembro 1991
- A Força de Trabalho e os Setores de Atividade - Uma Análise da Região Metropolitana de São Paulo - 1986-1990 - **nº 55**, outubro 1991
- Revisão da PNAD: Apuração das Informações: Módulo III do Anteprojeto - **nº 56**, novembro 1991
- Novos Usos para Pesquisa Industrial Mensal: A Evolução dos Salários Industriais, O Desempenho da Pecuária - **nº 57**, novembro 1991
- Revisão da PNAD: A Disseminação das Informações Módulo IV do Anteprojeto - **nº 58**, dezembro 1991
- Estatísticas Agropecuárias : Sugestões para o Novo Plano Geral de Informações - **nº 59**, dezembro 1991
- Análise Conjuntural e Pesquisa Industrial - **nº 60**, janeiro 1992
- Exploração dos Dados da Pesquisa Industrial Mensal de Dados Gerais - **nº 61**, fevereiro 1992

- Uma Proposta de Metodologia para a Expansão da Amostra do Censo Demográfico de 1991 - **nº 62**, outubro 1993
- Expansão da Fronteira e Progresso Técnico no Crescimento Agrícola Recente - **nº 63**, novembro 1993
- Avaliação das Condições de Habitação com Base nos Dados da PNAD - **nº 64**, setembro 1993
- Análise da Taxa de Desemprego Feminino no Brasil – **nº 65**, dezembro 1993
- Aspectos da Metropolização Brasileira: Comentários sobre os Resultados Preliminares do Censo Demográfico de 1991- **nº 66**, janeiro 1994
- Estimativas Preliminares de Fecundidade Considerando os Censos Demográficos, Pesquisas por amostragem e o Registro Civil - **nº 67**, janeiro 1994
- Apuração de Dados no IBGE: Problemas e Perspectivas - **nº 68**, fevereiro 1994
- Limeira - SP: Estimativas de Fecundidade e Mortalidade 1980/1988 - **nº 69**, março 1994
- Desemprego - Uma Abordagem Conceitual - **nº 70**, abril 1994
- Apuração dos Dados Investigados no Questionário Básico (CD 1.01) do Censo Demográfico de 1991 - **nº 71**, outubro de 1994
- Deslocamento Populacional e Segregação Sócio-Espacial – Migrantes Originários do Rio de Janeiro - **nº 72**, novembro de 1994
- Projeção Preliminar da População do Brasil para o Período 1980-2020 - **nº 73**, dezembro de 1994
- Considerações Preliminares Sobre a Migração Internacional no Brasil - nº **74**, janeiro de 1995
- Estatísticas Agropecuárias Censitárias no Âmbito do Mercosul - Brasil, Argentina e Uruguai - **nº 75**, julho de 1995
- Projeções Preliminares das Populações das Grandes Regiões para o Período 1991-2010 - **nº 76**, agosto de 1995
- Dinâmica da Estrutura Familiar no Sudeste Metropolitano, Chefia Feminina e Indicadores Sócio-Demográficos: Um exercício exploratório utilizando modelo da regressão múltipla - **nº 77**, setembro de 1995
- O Uso das Matrizes de Insumo-Produto e Matrizes de Inovação para Medir Mudanças Técnicas - **nº 78**, outubro de 1995
- Estimativas dos Fatores de Correção para o Registro de Nascimentos Utilizando Registros tardios a nível de Brasil, Grandes Regiões, Unidades da Federação e Regiões Metropolitanas 1974/1994 - **nº 79**, abril de 1996
- Aspectos de Amostragem Relativos ao Censo Cadastro de 1995 - **nº 80**, junho de 1996
- Tendências Populacionais no Brasil e Pressão Sobre o Mercado de Trabalho Futuro **- nº 81**, setembro de 1996
- Transformações Estruturais e Sistemas Estatísticos Nacionais - **nº 82**, setembro de 1996
- Metodologias para o Cálculo de Coeficientes Técnicos Diretos em um Modelo de Insumo-Produto - **nº 83**, outubro de 1996
- Avaliação da Cobertura da Coleta do Censo Demográfico de 1991 - **nº 84**, outubro de 1996
- Componentes da Dinâmica Demográfica Brasileira: Textos Selecionados - **nº 85**, novembro de 1996
- Apuração dos Dados Investigados pelo Questionário da Amostra - CD 1.02 do Censo Demográfico de 1991 - **nº 86**, dezembro de 1996
- Estudo Preliminar da Evolução dos Nascimentos, Casamentos e Óbitos 1974-1990 **- nº 87**, janeiro de 1997

- Sistema de Contas Nacionais - Tabelas de Recursos e Usos - Metodologia - **nº 88**, dezembro de 1997
- Aspectos de Amostragem da Pesquisa de Economia Informal Urbana 97 - **nº 89**, junho de 1998
- Comparações da Renda Investigada nos Questionários do Censo Demográfico de 1991 - **nº 90**, julho de 1998
- Uma Revisão dos Principais Aspectos dos Planos Amostrais das Pesquisas Domiciliares Realizadas pelo IBGE - **nº 91**, setembro de 1998
- Planejamento Amostral para as Pesquisas Anuais da Indústria e do Comércio - **nº 92**, outubro de 1998
- Aspectos de Amostragem da Pesquisa de Orçamentos Familiares 1995-1996 - **nº 93**, dezembro de 1998
- Reflexões sobre um Programa de Estatísticas Ambientais - **nº 94**, abril de 1999
- O Comportamento das Importações e Exportações Brasileiras com Base no Sistema de Contas Nacionais 1980 - 1997 (versão preliminar) - nº 95, maio de 1999
- Meio Ambiente: sua integração nos sistemas de informações estatísticas - nº 96, maio de 1999
- Conta da Terra: considerações sobre sua realização no Brasil - nº 97, dezembro de 1999

**Textos para discussão - nova série**

- **Número 1** - Sistema integrado de contas econômico-ambientais - SICEA : síntese e reflexões / Sandra De Carlo. - Rio de Janeiro : IBGE, Departamento de Contas Nacionais, 2000.
- **Número 2** - Aspectos da produção de informação estatística oficial no contexto da sociedade atual : algumas questões teórico-metodológicas / Rosa Maria Porcaro - Rio de Janeiro : IBGE, Departamento de Metodologia, 2000
- **Número 3** - A Cor denominada : um estudo do suplemento da Pesquisa Mensal de Emprego de julho/98 / José Luis Petruccelli. - Rio de Janeiro : IBGE, Departamento de População e Indicadores Sociais, 2000.
- **Número 4** - Indicadores para a agropecuária - Rio de Janeiro : IBGE, Departamento de Agropecuária, 2001.
- **Número 5** - Estudos para definição da amostra da Pesquisa Industrial Mensal de Emprego e Salário / Ana Maria Lima de Farias. - Rio de Janeiro : IBGE, Departamento de Indústria, 2001.
- **Número 6** - A declaração de cor/raça no censo 2000: um estudo comparativo / José Luis Petruccelli. - Rio de Janeiro : IBGE, Departamento de População e Indicadores Sociais, 2002..
- **Número 7** - Dimensões preliminares da responsabilidade feminina pelos domicílios: um estudo do fenômeno a partir dos censos demográficos 1991 e 2000 / Sonia Oliveira, Ana Lucia Sabóia, Bárbara Cobo - Rio de Janeiro : IBGE, Departamento de População e Indicadores Sociais, 2002.
- **Número 8** - Principais Aspectos de Amostragem das Pesquisas Domiciliares do IBGE - revisão 2002 / Zélia Magalhães Bianchini e Sônia Albieri - Rio de Janeiro : IBGE, Departamento de Metodologia, 2003.
- **Número 9** - Censo Demográfico 2000 - Resultados da Pesquisa de Avaliação da Cobertura da Coleta / Luís Carlos de Souza Oliveira, Marcos Paulo Soares de Freitas, Márcia Regina Martins Lima Dias, Cláudia Maria Ferreira Nascimento, Edie da Silva Mattos e João José Amado Ramalho Júnior - Rio de Janeiro : IBGE, Coordenação Técnica do Censo Demográfico, 2003.
- **Número 10** - Sistema de informação estatística e a sociedade da informação / Rosa Maria Porcaro - Rio de Janeiro : IBGE, Departamento de Metodologia, 2003.

- **Número 11 -** Indicadores para a agropecuária - 1996 a 2001 /Julio César Perruso, Marcelo de Moraes, Duriez, Roberto Augusto Soares P. Duarte e Carlos Alfredo Barreto Guedes - Rio de Janeiro : IBGE, Coordenação de Agropecuária, 2003.
- **Número 12** - A Unidade de Metodologia e a Evolução do Uso de Amostragem no IBGE, 2003 / Sônia Albieri - Rio de Janeiro : IBGE, Coordenação de Métodos e Qualidade, 2003.
- **Número 13** - Estimando a Precisão das Estimativas das Taxas de Mortalidade Obtidas a Partir da PNAD / Pedro Luis do Nascimento Silva e Djalma Galvão Carneiro Pessoa. - Rio de Janeiro : IBGE, Coordenação de Métodos e Qualidade, 2004.
- **Número 14** - A Qualidade na Produção de Estatísticas no IBGE / Zélia Magalhães Bianchini. - Rio de Janeiro : IBGE, Diretoria de Pesquisas, 2004
- **Número 15** - Calibration Estimation: When and Why, How Much and How / Pedro Luis do Nascimento Silva . - Rio de Janeiro : IBGE, Coordenação de Métodos e Qualidade, 2004
- **Número 16** - Um panorama recente da desigualdade no Brasil a partir dos dados da PNAD 2002 / Ana Lucia Saboia e Barbara Cobo. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2004
- **Número 17** – Processamento das Áreas de Expansão e Disseminação da Amostra no Censo Demográfico 2000 / Ari Nascimento Silva, Luiz Alberto Matzenbacher e Bruno Freitas Cortez. - Rio de Janeiro : IBGE, Coordenação de Métodos e Qualidade, 2004
- **Número 18** – Fatores de correção para o registro de nascimentos utilizando registros tardios segundo os grupos de idades das mulheres - Brasil e Unidades da Federação - 1984-2001 / Fernando Roberto Pires de Carvalho e Albuquerque e Selma Regina dos Santos. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2004
- **Número 19** – O processo de Imputação dos quesitos de migração no Censo Demográfico 2000 / Fernando Roberto P. de C. e Albuquerque, Janaína Reis Xavier Senna e Antonio Roberto Pereira Garcez - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2004
- **Número 20** – Tábuas de Mortalidade por sexo e grupos de idade - Grandes Regiões e Unidades da Federação - 1980, 1991 e 2000 / Fernando Roberto P. de C. e Albuquerque e Janaína Reis Xavier Senna - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2005
- **Número 21** – Tempo, trabalho e afazeres domésticos: um estudo com base nos dados da Pesquisa Nacional por Amostra de Domicílios - 2001 e 2005/ Cristiane Soares e Ana Lucia Saboia - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2007
- **Número 22** – Estimação de Intervalos de Confiança para Estimadores de Diferenças Temporais na Pesquisa Mensal de Emprego / Mauricio Franca Lila e Marcos Paulo soares de Freitas - Rio de Janeiro: IBGE, Coordenação de Trabalho e Rendimento e Coordenação de Métodos e Qualidade, 2007
- **Número 23** – Amostra Mestra para o Sistema Integrado de Pesquisas Domiciliares / Marcos Paulo Soares de Freitas, Maurício Franca Lila, Rosemary Vallejo de Azevedo e Giuseppe de Abreu Antonaci - Rio de Janeiro: IBGE, Coordenação de Métodos e Qualidade, 2007
- **Número 24** – Sistema Integrado de Pesquisas Domiciliares - SIPD / Coordenação de Trabalho e Rendimento - Rio de Janeiro: IBGE, 2007
- **Número 25** – Pesquisas Agropecuárias por Amostragem Probabilística no IBGE: Histórico e Perspectivas Futuras / Coordenação de Agropecuária - Rio de Janeiro: IBGE, 2007

- **Número 26** – Migração Pendular Intrametropolitana no Rio de Janeiro: Reflexões sobre o seu estudo, a partir dos Censos Demográficos de 1980 e 2000 / Antonio de Ponte Jardim e Leila Ervatti - Rio de Janeiro: IBGE, Coordenação de População e Indicadores Sociais, 2007
- **Número 27** – Características da fecundidade e da mortalidade segundo a condição migratória das mulheres, com base no quesito de "data fixa" / Fernando Roberto Pires de Carvalho e Albuquerque, Isabel Cristina Maria da Costa e Antonio Roberto Pereira Garcez - Rio de Janeiro: IBGE, Coordenação de População e Indicadores Sociais, 2007
- **Número 28** – Utilização de Modelos para Estimar a Mortalidade Brasileira nas Idades Avançadas / Jorcely Victório Franco, Juarez de Castro Oliveira e Fernando Roberto Pires de C. e Albuquerque - Rio de Janeiro: IBGE, Coordenação de População e Indicadores Sociais, 2007
- **Número 29** – Influência da mortalidade nos níveis de fecundidade da população brasileira e o intervalo médio entre duas gerações sucessivas - 1980, 1991, 2000 e 2005/ Fernando Roberto Pires de C. e Albuquerque e Maria lúcia Pereira do Nascimento - Rio de Janeiro: IBGE, Coordenação de População e Indicadores Sociais, 2008
- **Número 30** - Família nas pesquisas domiciliares : questões e propostas alternativas / Rosa Ribeiro, Ana Lúcia Sabóia - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2008
- **Número 31** – Setor e Emprego Informal no Brasil - Análise dos resultados da nova série do Sistema de Contas Nacionais / João Hallak Neto, Katia Namir, Luciene Kozovitz, Sandra Rosa Pereira - Rio de Janeiro : IBGE, Coordenação de Contas Nacionais, 2008
- **Número 32** - Diferenciais de idade entre os casais nas famílias brasileiras / Cristiane Soares. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, 2008
- **Número 33** – Estudos de modalidades alternativas de censos demográficos : aspectos de amostragem / IBGE, Diretoria de Pesquisas, Grupo de Trabalho de Amostragem, Estimação e Acumulação de Informações. - Rio de Janeiro : IBGE, 2009.
- **Número 34** – O Acompanhamento Estatístico da Fabricação de Medicamentos na Indústria Farmacêutica Brasileira/ Marcus José de Oliveira Campos e Luiz Antônio Casemiro dos Santos. - Rio de Janeiro : IBGE, Diretoria de Pesquisas, 2009.
- **Número 35** – Áreas mínimas de Comparação / Weuber da Silva Carvalho, Gilson Flaeschen. - Rio de Janeiro : IBGE, Diretoria de Pesquisas, 2010.
- **Número 36** – Contabilizando a Sustentabilidade: principais abordagens / Frederico Barcellos, Paulo Gonzaga M. de Carvalho e Sandra De Carlo. - Rio de Janeiro : IBGE, Diretoria de Pesquisas, 2010.
- **Número 37** – Indicadores sobre Trabalho Decente: Uma contribuição para o debate da desigualdade de gênero / Cíntia Simões Agostinho e Ana Lucia Saboia. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, Diretoria de Pesquisas, 2011.
- **Número 38** – Reflexões sobre pesquisas longitudinais: uma contribuição à implementação do Sistema Integrado de Pesquisas Domiciliares / Leonardo Athias. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, Diretoria de Pesquisas, 2011.
- **Número 39** – Desafios e possibilidades sobre os novos arranjos familiares e a metodologia para identificação de família no Censo / Ana Lucia Saboia, Bárbara Cobo e Gilson Gonçalves Matos. - Rio de Janeiro : IBGE, Coordenação de População e Indicadores Sociais, Diretoria de Pesquisas, 2012.

📖 **Número 40** – Metodologia Estatística da Pesca: Pesca embarcada / Aristides Pereira Lima Green e Guilherme Guimarães Moreira. - Rio de Janeiro : IBGE, Coordenação de Agropecuária e Coordenação de Métodos e Qualiddade, Diretoria de Pesquisas, 2012.