cidacs
Centro de Integração de Dados e Conhecimentos para Saúde
FIOCRUZ

Tratamento e Preparação para linkage da base SINAN HANSENÍASE

www.cidacs.bahia.fiocruz.br | cidacs@bahia.fiocruz.br | +55 71 3176-2357
Parque Tecnológico da Bahia: Rua Mundo, 121, Tecnocentro, sala 315 – Trobogy
Salvador, Bahia - CEP.: 41.745-715

Após ser recebida, a base do SINAN (Sistema de Informação de Agravos de Notificação) ela foi copiada pelo o NPD (Núcleo de produção de dados) para o ambiente de produção de dados para ser manipulado para preparar para o linkage. Os arquivos originais são mantidos para fim de comparação e validação das manipulações por isso todo procedimento aqui dito são realizados em cimas de cópias da base.

Antes de entregar para os pesquisadores, é necessário que a base passe por alguns processos de pré-processamento para tratamento da mesma, tais processos estão listados abaixo:

1. Conversão para csv;
2. Criação do dicionário do CIDACS;
3. Padronização das variáveis;
4. Correção das variáveis string para linkage;
5. Extração para linkage;

Cada uma dessas etapas serão explicadas com mais detalhe a seguir.

Conversão para csv

A base do sinan hanseníase foi recebida no formato CSV, formato padrão das bases que o NPD manipula.

As bases administrativas geralmente existem problemas de codificação e inclusão de caracteres especiais o que impacta diretamente na qualidade do dado, portanto é necessário submeter as diversas transformações. A base é lida no RStudio (no linux), onde acontece a primeira transformação: substituição dos caracteres que podem interferir na quantidade de colunas em um arquivo csv (contra barra, vírgula e aspas duplas), remoção de acentuações e modificação da codificação de caracteres de utf-8 para ascii (padrão de codificação de caracteres que melhor se adapta a todos os softwares que lerão a base). Para isso, o seguinte comando é utilizado:

```
library(foreign)

base_path_originals="/home/npd/bases/sinan/hanseniase/originais/SINAN_HANS_2000-201
6_ORIGINAL/"
base_path_convertion = "/home/npd/bases/sinan/hanseniase/v1/conversao_csv/"
file_list = list.files(path = base_path_originals)

for (file in file_list) {
 df <- read.dbf(paste(base_path_originals, file, sep = ""))
 df <- as.data.frame(list(lapply(df, function(x) { gsub("\"", ".", x)})))
 df <- as.data.frame(list(lapply(df, function(x) { gsub(",", ".", x)})))
 df <- as.data.frame(list(lapply(df, function(x) { gsub("([\\])", ".", x)})))
 df <- as.data.frame(list(lapply(df, function(x) { gsub("\n", ".", x)})))
 df <- as.data.frame(list(lapply(df, function(x) { gsub("\r", ".", x)})))
convertion_path = paste(paste(base_path_convertion, substr(file, 1, nchar(file) -
4), sep = ""), ".csv", sep = "")
write.csv(df, convertion_path, col.names = TRUE, row.names = FALSE, na = "")
```

```
system2('sed',paste('-i
\'y/áÁàÀãÃâÂéÉêÊíÍóÓõÕôÔúÚüÜçÇ/aAaAaAaAeEeEiIoOoOoOuUuUcC/\'', onvertion_path))
iconv_path = paste(base_path_convertion, substr(file, 1, nchar(file) - 4), sep =
"")
convertion_path_ascii = paste(base_path_convertion, substr(file, 1, nchar(file) -
4), sep = "")
iconv_path = paste(convertion_path_ascii, "_ascii.csv", sep = "")
 system2('iconv', paste(paste('-c -f utf-8 -t ascii', convertion_path), paste(">",
iconv_path)))
}
```

Foi verificado que a base do SINAN continha outros tipo de separadores no qual o código em R não tinha mapeado, então foi elaborado um código em Python que remove alguns separadores do ascii como tabulação vertical, horizontal entre outros e algumas variáveis de controle que estava impactando na análise de variáveis.

```
base_path_convertion = "/home/npd/bases/sinan/hanseniase/v1/conversao_csv/"
for _file in base_path_convertion:
   path_file = base_path + 'converted' + _file
   for x in [i for i in range(1,32) if i != 10]:
       cmd = "sed -i 's/" + str(struct.pack('B',x)) + "//g'" + path_file
       os.system(cmd)
```

Criação do dicionário do CIDACS

Com a base convertida, inicia-se a análise. A primeira etapa do processo é o mapeamento das variáveis consiste em identificar os tipos das variáveis e os valores aceitos por cada uma delas. Essa verificação deve ser feita com o auxílio dos dicionários de variáveis disponibilizados no site do Ministério da Saúde.

Uma vez mapeada, cria-se um dicionário aplicando o padrão adotado pelo CIDACS para descrição das variáveis, este dicionário posteriormente fica disponível para todos. Os campos presentes nele são:

1 **Nome da Variável**

Nome presente no dicionário de dados disponibilizado no site do Ministério da Saúde.

2 **Descrição Original**

Descrição presente no dicionário de dados disponibilizado no site do Ministério da Saúde, quando a variável é categórica neste campo estará descrito todos os valores aceitos.

3 **Nome da Variável – CIDACS**

Este nome normalmente é o mesmo da variável já existente na base de dados, mudando apenas o padrão de formatação para todas as letras em minúsculo, exceto nos casos em que é uma variável criada pelo CIDACS, quando isso ocorrer haverá uma sinalização.

4 **Harmonização**

Costuma-se replicar a mesma informação do campo 'Descrição Original'. No caso de categóricas, foi alterado as strings de cada categoria para um padrão numérico começando de 1 (um) e adicionando alguns casos especiais mostrados a seguir.

- **0 –** Registros nulos;
- **88 –** Registros que foram marcados em alguma opção que signifique abster-se da resposta, não saber responder ou alguma opção que não está presente na lista;
- **99 –** Registros que possuem valores fora do padrão aceito pela variável.

5 Tipo

Após análise e conhecimento do dicionário disponibilizado pelo Ministério da Saúde é possível afirmar qual tipo de dado cada variável aceita. Normalmente, os tipos preenchidos são:

- **Long –** Para variáveis numéricas com mais de 13 dígitos;
- **Byte –** Para variáveis categóricas;
- **Integer –** Para variáveis inteiras com menos de 13 dígitos que não são categóricas;
- **Date –** Para variáveis de data;
- **String –** Para as variáveis com caracteres alfanuméricos.

6 Quantidade de Missings

Expõe em porcentagem a quantidade de valores nulos presentes em cada variável.

7 Comentários

Campo aberto para comentários em geral sobre a variável

Como resultado desta etapa temos um dicionário padronizado e pronto para consulta, seja para a harmonização das variáveis ou para os pesquisadores.

Padronização das variáveis

Para garantir que os dados presentes na base condizem com o que está no dicionário e para que o esquema a ser definido em *parquet* funcione, é preciso tratar as variáveis para só então assegurar que o tipo de cada uma delas esteja correto. Assim, é necessário realizar algumas correções antes da conversão para *parquet*. Este processo foi nomeado como **padronização**, e garante a consistências das variávei do tipo:

1 Data

Para um data ser considerada válida é necessário que tenha no mínimo 7 caracteres numéricos (no formato ddmmyyyy) cujos primeiros dois dígitos representem um dia válido (de 01 a 31), os dois dígitos seguintes representem um mês válido (de 01 a 12) e que os quatro últimos dígitos representam um ano válido (maior do que 1000[1]).

O tratamento realizado substitui por None (representação de um campo não preenchido) todos os registros encontrados em variáveis do tipo data que não se enquadrem nessas premissas.

2 Categórica

Com o auxílio do dicionário do CIDACS construído anteriormente, cria-se uma lista com as categorias válidas para cada variável e esta é submetida a uma função, juntamente com o nome da variável correspondente. Assim, todos os registros que se enquadram nas categorias corretas são mantidos, os registros nulos têm seu valor transformado para 0 (zero) , os inconsistentes são mapeados para 99 e os ignorados (caso sinalizado no dicionário) são transformados em 88.

[1] A ferramenta *Spark* não valida anos como 0001. Assim, para garantir a consistência do banco, do ponto de vista técnico, consideramos que os anos a partir de 1000 eram válidos.

3 Inteiro

Verifica-se a existência de caracteres que fujam do padrão numérico, e quando encontrados ocorre a substituição por *None*, pois estes não condizem com o tipo do campo.

Além desses três tipos, por padrão, todas as variáveis de sexo que não estão em formato numérico, foram modificadas. Assim, variáveis do tipo M e F foram transformadas em 1 e 2, respectivamente[2]. Além disso, todos os nomes de colunas foram padronizados para nomes minúsculos e todas variáveis que referenciavam uma tabela anexo foram verificadas a fim de garantir que os valores presentes nelas estavam em suas respectivas tabelas, caso contrário o mesmo tratamento feito para categóricas era aplicado.

Uma vez corrigidas as inconsistências relacionadas aos tipos de variáveis, a base de dados é salva numa nova versão para então ser definido o esquema em parquet.

Para definição do esquema utilizamos os seguintes tipos de variáveis:

1. LongType: para variáveis numéricas com mais de 13 dígitos
2. ByteType: para variáveis categóricas ou numéricas bem pequenas
3. IntegerType: para variáveis inteiras com menos de 13 dígitos que não são categóricas
4. DateType: para variáveis de data
5. StringType: para as variáveis com caracteres alfanuméricos
6. DoubleType: para variáveis com ponto decimal

Os esquemas devem ser definidos na mesma ordem em que as variáveis aparecem no banco. Uma vez definido o esquema, ele é aplicado e a

[2] Essa transformação foi realizada pensando nos processos de *linkage* com o *baseline*, visto que no *baseline* a variável sexo segue esse mesmo formato.

base é salva em formato *parquet*, finalizando a etapa de conversão. Para visualizar o script de aplicação do esquema em *parquet*, basta clicar aqui.

Correção das variáveis de String para linkage

Para o processo de linkage, todas as variáveis utilizadas (geralmente: nome, nome da mãe, data de nascimento, código de município de residência e sexo) precisam estar tratadas. Dentre elas, as variáveis de data de nascimento, sexo e município, por se tratar de inteiro, categórica ou data, recebem tratamento no momento da conversão das bases de csv para parquet, restando apenas nome e nome da mãe para ser validadas.

Para as duas variáveis restante (nome e nome da mãe) o tratamento é mais complexo, pois se trata de variáveis do tipo String que possui inconsistências, como erros de digitação e também erros de codificação que levam a Inelegibilidade dos nomes, tais erros são ocasionados por equívocos na digitação e também conversão das bases de dados podendo surgir problemas de codificação. Inicialmente, faz-se uma verificação dos tipo de inconsistência que ocorre na base. Para isso é criado um banco de nomes e sobrenomes de três bases de dados, CADU (Cadastro único), SINASC (Sistema de Informação sobre Nascidos Vivos) e SIM (Sistema de Mortalidade) esse banco legitima se um nome ou sobrenome do SINAN é válido ou não. As inconsistências associadas a esta base foram:

1. **Erros ortográficas ou caracteres inválidos:** Nesses casos, algumas vezes a letra era trocada por número, ou havia uma sequência de letras, caracteres especiais e números que não traziam significado para a variável. Como: DARHKUX, S., EB$....UKNXC, #;ILVA

2. **Erros por abreviações em excesso:** Alguns nomes contém abreviações no sobrenome como por exemplo .S .M, com isso esses nomes não são significativos porém na correção é possível recuperá-lo.

Uma vez mapeado os erros, é preciso definir os procedimentos de tratamento para esse tipo de campo. Com o banco de nomes válidos construído foi possível criar um novo banco de dados do erros catalogados durante a etapa de verificação de inconsistência que estão presentes na base de SINAN-HANS.

A variável nome e nome da mãe continham mais ou menos 5% e 7% de erros respectivamente, tais erros são totalmente inelegíveis, portanto foi necessário utilizar também a variável fonética (concatenação do primeiro nome e o último nome do indivíduo) para recuperar os nomes corrigindo os erros. A correção foi efetuada construindo um dicionário de correção, que contém os erros e o significado deste erro. Por exemplo:

Erro	Significado
#!<JMWB	SILVA
$..EQ=VC	COSTA
$..NK7C	LIMA

Após a recuperação do nomes e sobrenomes, caso ainda exista algum nome que não exista no banco de nomes do CADU, SIM, SINASC tais nomes serão removidos.

5. Extração para linkage

Além do banco gerado para avaliação individual, há necessidade de extração das variáveis para o *linkage*. Assim, apenas as variáveis, nome do paciente, nome da mãe, data de nascimento, sexo e código do município de residência, idade, raça e código do sus são extraídas da base.

Desta forma, tem-se duas bases distintas, a base decorrente da finalização da etapa de padronização das variáveis e a base extraída para *linkage*, decorrente dos procedimentos citados na etapa 4 (Correção das variáveis strings para linkage).

.