Extração, Tarnformação e Carga (ETL) de dados do SUS via Protocolo de Transferência de Arquivos (FTP)

Bibliotecas do R

As operações a seguir baixam e carregam os pacotes necessários o para processamento dos dados.

```
if(!require(RCurl)) {install.packages("RCurl"); require(RCurl)}
# funcao getURL

if(!require(downloader)) {install.packages("downloader"); require(downloader)}
# funcao download

if(!require(stringr)) {install.packages("stringr"); require(stringr)}
# lpad str_pad

if(!require(gsubfn)) {install.packages("gsubfn"); require(gsubfn)}
```

```
Loading required package: read.dbc

Warning message in library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
“there is no package called ‘read.dbc’”
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘read.dbc’ is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages”
Loading required package: read.dbc

Warning message in library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
“there is no package called ‘read.dbc’”
```

pacote read.dbc

> Saiba mais sobre o read.dbc em https://pt.linkedin.com/pulse/datasus-conhe%C3%A7a-nova-ferramenta-para-ler-arquivos-dbc-petruzalek

```
if(!require(read.dbc)) {devtools::install_github("danicat/read.dbc"); require(read.dbc)}
# le arquivo DBC da estrategia tabnet/tabwin de disseminacao
```

```
Downloading GitHub repo danicat/read.dbc@HEAD

── R CMD build ─────────────────────────────────────────────────────────
* checking for file ‘/tmp/RtmpvSL86P/remotesdc15ee3160/danicat-read.dbc-eb654e5/DESCRIPTION’ ... OK
* preparing ‘read.dbc’:
* checking DESCRIPTION meta-information ... OK
* cleaning src
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
Omitted ‘LazyData’ from DESCRIPTION
* building ‘read.dbc_1.0.5.tar.gz’

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Loading required package: read.dbc
```

Parâmetros

A título de exemplo vamos trabalhar com o Protocolo Clínico e Diretriz Terapêutica **Espondilite Ancilosante**, disponível em https://www.gov.br/conitec/pt-br/midias/protocolos/20210428_pcdt-espondilite-ancilosante-1.pdf .

```
cid10=c("M45", "M468") # Espondilite Ancilosante
sigtap=c(
  "0601010019", # ADALIMUMABE (A) 40 MG INJETAVEL- SERINGA PREENCHIDA (POR TRATAMENTO MENSAL)   Revogado desde 06/2010
  "0604380011", # ADALIMUMABE 40 MG INJETAVEL (POR SERINGA PREENCHIDA)
  "0604380062", # ADALIMUMABE 40 MG INJETÁVEL (POR SERINGA PREENCHIDA)
  "0604380097", # ADALIMUMABE 40 MG INJETÁVEL (FRASCO AMPOLA)
  "0604380127", # ADALIMUMABE 40 MG INJETÁVEL ( POR SERINGA PREENCHIDA)( BIOSSIMILAR A)
  "0604380135", # ADALIMUMABE 40 MG INJETÁVEL (POR SERINGA PREENCHIDA) (BIOSSIMILAR B)
  "0601010027", # ETANERCEPTE (A)25 MG INJETAVEL -FRASCO-AMPOLA (POR TRATAMENTO MENSAL) Revogado desde 06/2010
```

```
  "0601010051", # ETANERCEPTE 50MG INJETAVEL- FRASCO AMPOLA (POR TRATAMENTO MENSAL) Revogado desde 06/2010
  "0604380020", # ETANERCEPTE 25 MG INJETÁVEL (POR FRASCO-AMPOLA OU SERINGA PREENCHIDA)
  "0604380038", # ETANERCEPTE 50MG INJETAVEL (POR FRASCO-AMPOLA OU SERINGA PREENCHIDA)(ORIGINADOR)
  "0604380100", # ETANERCEPTE 50 MG INJETÁVEL (POR FRASCO-AMPOLA OU SERINGA PREENCHIDA)(BIOSSIMILAR A)
  "0601010035", # INFLIXIMABE (A)10 MG/ML 10 ML INJETAVEL (FRASCO-AMPOLA- POR TRATAMENTO MENSAL)    Revogado desde 06/2010
  "0601010043", # INFLIXIMABE 10 MG/ML INJETAVEL (POR FRASCO-AMPOLA 10 ML)  Revogado desde 06/2010
  "0604380046", # INFLIXIMABE 10 MG/ML INJETAVEL (POR FRASCO-AMPOLA COM 10 ML)
  "0604380054", # INFLIXIMABE 10 MG/ML INJETAVEL (POR FRASCO-AMPOLA COM 10 ML)
  "0604380119", # INFLIXIMABE 10 MG /ML INJETÁVEL (POR FRASCO-AMPOLA COM 10 ML) (BIOSSIMILAR A)
  "0604380089", # GOLIMUMABE 50 MG INJETÁVEL (POR SERINGA PREENCHIDA
  "0604380070", # CERTOLIZUMABE PEGOL 200 MG/ML INJETÁVEL (POR SERINGA PREENCHIDA)
  "0604690029" # SECUQUINUMABE 150 MG/ML SOLUÇÃO INJETÁVEL (POR SERINGA PREENCHIDA)
) # biologicos


ufs=c('AC', 'AM', 'AP', 'PA', 'RO', 'RR', 'TO',
      'AL', 'BA', 'CE', 'MA', 'PB', 'PE', 'PI', 'RN', 'SE',
      'ES', 'MG', 'RJ', 'SP', 'PR', 'RS', 'SC', 'DF', 'GO', 'MS', 'MT')

ano=18:22
mes=str_pad(1:12, 2, pad="0")

url="ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/"
```

Estrutura dos dados

Criar tabelas vazias no formato **data.frame** contendo as variáveis desejadas.

O dicionário de dados do Sistema de Informações Ambulatoriais está disponível em ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Doc/.

Como baixar o dicionário de dados

> **Alternativa 1**
>
> Os enderelos do tipo ftp:// usualmente não funcionam no navegador de internet. Cole o endereço no navegador de arquivos.

> **Alternativa 2**
>
> Baixe usando o R em sua máquina local com a função `download.file`.

```
# R
download.file(
  "ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Doc/Informe_Tecnico_SIASUS_2019_07.pdf",
  destfile = "Informe_Tecnico_SIASUS_2019_07.pdf"
)
```

```
 paam_estrutura=data.frame(
  PA_AUTORIZ = numeric(),
  PA_CMP = numeric(),
  PA_MVM = numeric(),
  PA_CIDPRI = character(),
  PA_CIDSEC = character(),
  PA_PROC_ID = character(),
  PA_QTDAPR = numeric(),
  PA_SEXO = character(),
  PA_IDADE = numeric(),
  PA_MUNPCN = numeric(),
  uf_processamento  = character(),
  AP_CNSPCN = character()
 )

 pa_estrutura=data.frame(
  PA_AUTORIZ = numeric(),
  PA_CMP = numeric(),
  PA_MVM = numeric(),
  PA_CIDPRI = character(),
  PA_CIDSEC = character(),
  PA_PROC_ID = character(),
  PA_QTDAPR = numeric(),
  PA_SEXO = character(),
  PA_IDADE = numeric(),
  PA_MUNPCN = numeric(),
  uf_processamento  = character()
 )

 am_estrutura=data.frame(
  AP_AUTORIZ = numeric(),
  AP_PRIPAL = character(),
```

```
    AP_CIDPRI = character(),
    AP_CNSPCN = character()
  )
```

Exemplo de dado

O Sistema de Informação Ambulatorial (SIA) apreenta vários subsistemas, a saber:

- **PA Produção Ambulatorial**
- AB Laudo de Acompanhamento à Cirurgia Bariátrica
- ABO Acompanhamento Pós Cirurgia Bariátrica
- ACF Laudo de Confecção de Fístula
- AD Laudos Diversos
- **AM Laudo de Medicamentos**
- AMP Laudo de Acompanhamento Multiprofissional
- AN Laudo de Nefrologia
- AQ Laudo de Quimioterapia
- AR Laudo de Radioterapia
- ATD Laudo de Tratamento Dialítico
- BI Boletim Individual

Os arquivos estão no formado DBF, cujo nome é padronizado:

prefixo | UF | ano com dois dígitos | mês com dois dídigos.

Exemplos: PARR2301.dbc, AMPR2201.dbc, AQBA1801.dbc.

Vamos usar o `download.file` e o `read.dbc` para ler um arquivo direto do diretório FTP.

```
 download.file(
  paste0(url,'PAAC2212.dbc'),
  destfile = "arquivo.dbc"
)
paac2212=read.dbc("arquivo.dbc")
head(paac2212)
```

	PA_CODUNI	PA_GESTAO	PA_CONDIC	PA_UFMUN	PA_REGCT	PA_INCOUT	PA_INCURG
	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
1	2001586	120000	EP	120040	0000	0000	0000
2	5336171	120000	EP	120020	0000	0000	0000
3	0128619	120000	EP	120040	0000	0000	0000
4	2000393	120000	EP	120070	0000	0000	0000
5	2000970	120000	EP	120034	0000	0000	0000
6	2001063	120000	EP	120040	0000	0000	0000

Lista de arquivos DBC

A lista completa dos arquivos dbc do SIA é obtida com a função `getURL`.

```
url="ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/"

# lista arquivos dbc do diretorio FTP
aux <-
  getURL(
    url,
    verbose = TRUE,
    ftp.use.epsv = FALSE,
    dirlistonly = TRUE,
    crlf = TRUE
  )
listadbc=strsplit(aux, "\r*\n")[[1]]

length(listadbc) # numero de arquivos

listadbc[sample(1:length(listadbc),20, replace = TRUE)]
# amostra de arquivos DBC
```

```
43158
'PSRJ1711.dbc' · 'AQMT2010.dbc' · 'BIPR1408.dbc' · 'ARMS1907.dbc' · 'AMRR1304.dbc'
```

Lista de UF por mês de competência

O controle da ordem de manipulação dos arquivos segundo estado e mês é fundamental na carga.

A função `expand.grid` é empregada para gerar as combinações de UF, ano e mês que integram o nome dos arquivos a serem manipulados.

```
uf='GO'
aux=as.matrix(expand.grid(uf,ano,mes))
ufaamm=sort(paste0(
  aux[,1],
  aux[,2],
  aux[,3]
))
ufaamm
```

```
'GO1801' · 'GO1802' · 'GO1803' · 'GO1804' · 'GO1805' · 'GO1806' · 'GO1807' · 'GO1808' ·
'GO1809' · 'GO1810' · 'GO1811' · 'GO1812' · 'GO1901' · 'GO1902' · 'GO1903' · 'GO1904' ·
'GO1905' · 'GO1906' · 'GO1907' · 'GO1908' · 'GO1909' · 'GO1910' · 'GO1911' · 'GO1912' ·
'GO2001' · 'GO2002' · 'GO2003' · 'GO2004' · 'GO2005' · 'GO2006' · 'GO2007' · 'GO2008' ·
```

Enriquecimento do subsistema principal do SIA

O subsistema principal do **SIA** é chamado de corpo da Autorização de Procedimentos Ambulatoriais (antiga sigla APAC) e são disseminados sob o prefixo **PA**.

Por meio do número de autorização, em geral, **PA_AUTORIZ** ou **AP_AUTORIZ**, é possível completar o arquivo principal.

A tarefa é fundamental para computar o número de usuários com quantidade aprovada de dado procedimento, sobretudo a partir do arquivo de medicamentos de prefixo **AM**.

Veja o exemplo ilustrado para o primeiro valor no índice do vetor `ufaamm`.

Processamento dos arquivos PA da mesma UF e mês

Eventualmente arquivos do mês mês que ultrapassam cerca de dois milhões de registros são fragmentados e mais de um arquivo, por exemplo, **PASP2301a.dbc**, **PASP2301b.dbc** e **PASP2301c.dbc**.

Por isso é utilizada a estrutura de repetição `for`.

Note que os filtros de **CID10** e procedimento **SIGTAP** são usados logo após a carga para reduzir a ocupação na memória e, portanto, o recurso computacional necessário.

Foram estabelecidos filtros segundo a doença selecionada em `PA_CIDPRI %in% cid10` e procedimento em `PA_PROC_ID %in% sigtap`.

Adicionalmente, o vetor de atributos foi aplicado a fim de desprezar os demais.

```
c("PA_AUTORIZ","PA_CMP","PA_MVM" ,"PA_CIDPRI", "PA_CIDSEC",
  "PA_PROC_ID", "PA_QTDAPR","PA_SEXO", "PA_IDADE", "PA_MUNPCN"
)
```

> **Nota:** É recomendável baixar previamente os arquivo DBC para o disco local Assim, basta substituir a linha de comando
>
> `download.file(paste0(url,listadbc_pa[j]), destfile = "arquivo.dbc")`
>
> por
>
> `arquivo=paste0(dirdbc,listadbc_pa[j])`

```
i=1
ufaamm[i]

# filtra apenas arquivos contendo o respectivo UFAAMM
listadbc_pa=subset(
      listadbc,
      grepl(paste0("^PA",ufaamm[i]), listadbc)
    )
listadbc_pa

pa=pa_estrutura

# incorpora o arquivo PA
    for (j in 1:length(listadbc_pa)) {
      arquivo = "arquivo.dbc"
      download.file(paste0(url,listadbc_pa[j]), destfile = arquivo)
      # arquivo=paste0(dirdbc,listadbc_pa[j])
      aux=subset(
        read.dbc(arquivo)[,c(
          "PA_AUTORIZ","PA_CMP","PA_MVM" ,"PA_CIDPRI", "PA_CIDSEC",
          "PA_PROC_ID", "PA_QTDAPR","PA_SEXO", "PA_IDADE", "PA_MUNPCN"
        )],
```

```
      PA_CIDPRI %in% cid10 &
        PA_PROC_ID %in% sigtap
    )

    if (nrow(aux)==0) {
      aux=pa
    } else {
      aux$uf_processamento = uf
      pa=rbind(pa,aux)
    }

  }
head(pa)
```

```
'GO1801'
'PAGO1801.dbc'
```

A data.frame: 6 × 11

	PA_AUTORIZ	PA_CMP	PA_MVM	PA_CIDPRI	PA_CIDSEC	PA_PROC_ID	PA_QTDA
	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<ir
1975	5217202829366	201801	201801	M45	0000	0604380011	
2015	5218200283483	201801	201801	M45	0000	0604380054	
3967	5217203323300	201801	201801	M45	0000	0604380011	
3969	5217203318943	201801	201801	M45	0000	0604380011	
3971	5218200003445	201801	201801	M45	0000	0604380011	
3975	5217203358136	201801	201801	M45	0000	0604380011	

Processamento do arquivo AM por UF e mês

Analogamente ao que ocorreu com o arquivo **PA**, é processado o arquivo **AM**.

```
listadbc_am=subset(
  listadbc,
  grepl(paste0("^AM",ufaamm[i]), listadbc)
)

   for (j in 1:length(listadbc_am)) {
    arquivo = "arquivo.dbc"
    download.file(paste0(url,listadbc_am[j]), destfile = arquivo)
    # arquivo=paste0(dirdbc,listadbc_am[j])
    if (file.exists(arquivo)) {
      aux=subset(
        read.dbc(arquivo)[,c(
          "AP_AUTORIZ", "AP_PRIPAL",
          "AP_CIDPRI",  "AP_CNSPCN"
        )],
        AP_CIDPRI %in% cid10 & AP_PRIPAL %in% sigtap
      )
      am=rbind(am,aux)
    }
  }
head(am)
```

```
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
ERROR while rich displaying an object: Error in gsub(chr, html_specials[[chr]

Traceback:
1. tryCatch(withCallingHandlers({
 .     if (!mime %in% names(repr::mime2repr))
 .         stop("No repr_* for mimetype ", mime, " in repr::mime2repr")
 .     rpr <- repr::mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler), error = outer_handler)
2. tryCatchList(expr, classes, parentenv, handlers)
3. tryCatchOne(expr, names, parentenv, handlers[[1L]])
4. doTryCatch(return(expr), name, parentenv, handler)
5. withCallingHandlers({
 .     if (!mime %in% names(repr::mime2repr))
 .         stop("No repr_* for mimetype ", mime, " in repr::mime2repr")
 .     rpr <- repr::mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler)
6. repr::mime2repr[[mime]](obj)
7. repr_markdown.data.frame(obj)
8. repr_matrix_generic(obj, "\n%s\n\n%s%s\n", sprintf("|%%s\n|%s|\n",
 .     underline), NULL, " <!--/--> |", " %s |", "%s", "|%s\n",
 .     " %s |", " %s |", escape_fun = markdown_escape, rows = rows,
 .     cols = cols, ...)
9. lapply(seq_len(nrow(x)), function(r) {
 .     row <- escape_fun(slice_row(x, r))
 .     cells <- sprintf(cell, row)
 .     if (has_rownames) {
 .         row_head <- sprintf(row_head, escape_fun(rownames(x)[[r]]))
 .         cells <- c(row_head, cells)
 .     }
 .     sprintf(row_wrap, paste(cells, collapse = ""))
 . })
10. FUN(X[[i]], ...)
11. escape_fun(slice_row(x, r))
12. html_escape(values, do_spaces = FALSE)
13. gsub(chr, html_specials[[chr]], text, fixed = TRUE)
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
```

```
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
```

```
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 4 is invalid in this locale”
```

Substituição dos caracteres especias do CNS criptografado

A fim de agilizar o processamento, os caracteres especiais do **Cartão Nacional de Saúde** (CNS) criptografado são substituídos por números.

```
am2=unique(am[,c("AP_AUTORIZ","AP_CNSPCN")])

      am2$AP_CNSPCN=
        gsubfn(
          ".",
          list(
            "{" = "0", "}" = "9", "~" = "5",
            "\177" = "7", "Ç" = "6", "ä" = "8",
            "ü" = "4", "é" = "1", "|" = "2", "â" = "3"
          ),
          iconv(am2$AP_CNSPCN, "CP861", "UTF-8")
        )
head(am2)
```

A data.frame: 6 × 2

	AP_AUTORIZ	AP_CNSPCN
	<fct>	<chr>
344	5217203327765	109801621286415
345	5217203183050	100405717722145
346	5217203322628	383007002133371
351	5218200226162	383009000960458
352	5218200062010	383007087503755
353	5217202866777	104304919345396

Junção do PA e AM

A junção ocorre segundo código da autorização. Note a incorporação do atributo `AP_CNSPCN`. Se mais atributos forem necessários basta incorporar ao vetor do arquivo `AM`.

```
      paam = merge(
        pa, am2,
```

```
    by.x=c("PA_AUTORIZ"),
    by.y=c("AP_AUTORIZ"),
    all.x = TRUE
  )
head(paam)
```

A data.frame: 6 × 12

	PA_AUTORIZ	PA_CMP	PA_MVM	PA_CIDPRI	PA_CIDSEC	PA_PROC_ID	PA_QTDAPR	P
	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<int>	
1	5217202706364	201801	201801	M45	0000	0604380011	2	
2	5217202716407	201801	201801	M45	0000	0604380054	3	
3	5217202751365	201801	201801	M45	0000	0604380054	3	
4	5217202756403	201801	201801	M45	0000	0604380011	2	
5	5217202763003	201801	201801	M45	0000	0604380054	0	
6	5217202763400	201801	201801	M45	0000	0604380011	2	