

# Sabeis

Sala Aberta de Inteligência em Saúde

# Curso RebrATS 2025

Acesso aos Dados da SABEIS via FTP

utilizando R

Felipe Ferré e Amanda Lyrio

05/11/2025

# Sobre o Curso

# Oferta

**Curso:** Acesso aos Dados da SABEIS via FTP utilizando a Linguagem R

**Grupo Elaborador:** Departamento de Gestão e Incorporação de Tecnologias em Saúde (DGITS/SECTICS/MS)

**Docentes:** Felipe Ferré e Amanda Lyrio

**Monitores:** Mariá Pereira, Jéssica Barreto, Laís Lessa e Michael Ruberson

**Carga Horária:** 12 horas · 25 vagas

**Público-alvo:** Profissionais dos NATS com experiência prévia em R, interessados em ampliar autonomia e agilidade no acesso e tratamento de dados da SABEIS para subsidiar análises e estudos de ATS.

**Pré-requisito:** Teste prático para comprovar domínio básico de R.

**Material sugerido:** Apostila do curso.

# Atividade

- Conexão à SABEIS por FTP.
- Download, estruturação e leitura dos dados em R.
- Demanda aferida.
- Avaliação pós-incorporação.

## **Objetivo**

Capacitar participantes a acessar e manipular dados da SABEIS via FTP em R para fortalecer análises em ATS.

## **Material necessário**

- Computador com internet.
- R e RStudio instalados.

# Conteúdo programático

## **05/11 – manhã (8-12h)**

Fundamentos práticos dos dados ambulatoriais do SUS e organização do repositório SABELS.

## **05/11 – tarde (14-18h)**

ETL de dados tabulados via FTP e organização em repositórios estruturados (RData e SQL).

# Referências

- [Apresentação](https://github.com/sabeis-ats/curso_rebrats2025) ([https://github.com/sabeis-ats/curso\\_rebrats2025](https://github.com/sabeis-ats/curso_rebrats2025))
- FERRÉ, FELIPE; DE OLIVEIRA, GUSTAVO ; DE QUEIROZ, MARIANA ; GONÇALVES, FLÁVIO . Sala de Situação aberta com dados administrativos para gestão de Protocolos Clínicos e Diretrizes Terapêuticas de tecnologias providas pelo SUS. In: Anais Principais do Simpósio Brasileiro de Computação Aplicada à Saúde, 2020, Brasil. Anais Principais do Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2020). p. 392. [link](https://sol.sbc.org.br/index.php/sbcas/article/view/11530) (<https://sol.sbc.org.br/index.php/sbcas/article/view/11530>)
- Felipe Ferré. Modelagem e gestão de banco de dados com SQL e integração com o R. In: Erika Santos de Aragão; Luís Eugenio Portela Fernandes de Souza; Marcio Santos da Natividade; Vinícius de Araújo Mendes. (Org.). Avaliação de impacto das políticas de saúde: um guia para o SUS. 1ed.Brasília: Editora MS, 2023, v. , p. 1-715. [link](https://bvsms.saude.gov.br/bvs/publicacoes/avaliacao_impacto_politicas_saude_guia_sus.pdf) ([https://bvsms.saude.gov.br/bvs/publicacoes/avaliacao\\_impacto\\_politicas\\_saude\\_guia\\_sus.pdf](https://bvsms.saude.gov.br/bvs/publicacoes/avaliacao_impacto_politicas_saude_guia_sus.pdf))
- [Sabeis - ETL](https://rpubs.com/sabeis/etl) (<https://rpubs.com/sabeis/etl>)

## Tabelas:

[GitHub Sabeis - BD Geral](https://github.com/sabeis-ats/bd_geral/) ([https://github.com/sabeis-ats/bd\\_geral/](https://github.com/sabeis-ats/bd_geral/)).

# Introdução aos dados abertos do SUS e à SABEIS

# A Sala Aberta de Inteligência em Saúde (SABEIS)

- Metodologia de extração, transformação, carga e curadoria de dados.
- Iniciou em 2019 no âmbito do DGITS/SECTICS/MS (Secretaria Executiva da Conitec).
- Usa apenas software livre e dados abertos.
- É mantida por profissionais de saúde a serviço do SUS com tecnologia robusta e simples.
- Validada com dados oficiais a partir de marcadores de produção.
- Fonte principal: Ministério da Saúde - FTP de dados ambulatoriais do SUS.
- Foco em medicamentos do Componente Especializado da Assistência Farmacêutica.
- Prima em favorecer o diálogo com a sociedade civil e a gestão partindo de dados comuns e metodologia transparente.



# A estratégia de disseminação TabNet/TabWin

Acesse o [sítio do TabNet](https://datasus.saude.gov.br/informacoes-de-saude-tabnet/) (<https://datasus.saude.gov.br/informacoes-de-saude-tabnet/>) para conhecer o elenco de informações disponíveis segundo:

- Indicadores de Saúde e Pactuações
- Assistência à Saúde
- Epidemiológicas e Morbidade
- Rede Assistencial
- Estatísticas Vitais
- Demográficas e Socioeconômicas
- Inquéritos e Pesquisas
- Saúde Suplementar (ANS)
- Informações Financeiras
- Estatísticas de acesso ao TABNET

# TabNet - Assistência à Saúde

- [Produção Hospitalar \(SIH/SUS\)](https://datasus.saude.gov.br/aceso-a-informacao/producao-hospitalar-sih-sus/) (<https://datasus.saude.gov.br/aceso-a-informacao/producao-hospitalar-sih-sus/>)
- [Produção Ambulatorial \(SIA/SUS\)](https://datasus.saude.gov.br/aceso-a-informacao/producao-ambulatorial-sia-sus/) (<https://datasus.saude.gov.br/aceso-a-informacao/producao-ambulatorial-sia-sus/>)
- [Imunizações – desde 1994](https://datasus.saude.gov.br/aceso-a-informacao/imunizacoes-desde-1994/) (<https://datasus.saude.gov.br/aceso-a-informacao/imunizacoes-desde-1994/>)
- [Atenção Básica – Saúde da Família – de 1998 a 2015](https://datasus.saude.gov.br/atencao-basica-saude-da-familia-de-1998-a-2015/) (<https://datasus.saude.gov.br/atencao-basica-saude-da-familia-de-1998-a-2015/>)
- [Vigilância Alimentar e Nutricional](https://datasus.saude.gov.br/aceso-a-informacao/vigilancia-alimentar-e-nutricional/) (<https://datasus.saude.gov.br/aceso-a-informacao/vigilancia-alimentar-e-nutricional/>)

# TabNet - Produção Ambulatorial

- [Por local de atendimento - a partir de 2008](http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sia/cnv/qabr.def) (<http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sia/cnv/qabr.def>)
- Por local de atendimento - de 1994 a 2007
- Por local de residência - a partir de 2008
- Por gestor - a partir de 2008

# TabNet - Navegação

- Linha (p.ex., Região, UF, Procedimento, Ano/mês de processamento)
- Coluna (p.ex, Grupo de Procedimento, Esfera Jurídica)
- Conteúdo (Qtd. aprovada, Valor aprovado)
- Período
- Seleções disponíveis (filtros)

# FTP e repositórios públicos do SUS

O FTP (File Transfer Protocol) é um protocolo usado para transferir arquivos entre servidores e computadores.

No contexto do SUS (Sistema Único de Saúde), o FTP é utilizado como repositório público de dados, permitindo que pesquisadores, gestores e analistas baixem grandes bases de informações oficiais.

Sigla	Descrição	URL
FTP	Protocolo de Transferência de Arquivos, usado para mover arquivos entre servidores na internet. Repositório de dados públicos do SUS para download via FTP.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/">ftp://ftp.datasus.gov.br/dissemin/publicos/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/">ftp://ftp.datasus.gov.br/dissemin/publicos/</a> )
SIA	Sistema de Informações Ambulatoriais do SUS, gerencia dados sobre atendimentos ambulatoriais.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIASUS/200801_/Dados/</a> )
SIH	Sistema de Informações Hospitalares do SUS, registra dados de internações e procedimentos hospitalares.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIHSUS/200801_/Dados/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIHSUS/200801_/Dados/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIHSUS/200801_/Dados/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIHSUS/200801_/Dados/</a> )

---

# FTP e repositórios públicos do SUS

Sigla	Descrição	URL
SIM	Sistema de Informações sobre Mortalidade, coleta dados de óbitos registrados no Brasil.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIM/CID10/DORES/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIM/CID10/DORES/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SIM/CID10/DORES/">ftp://ftp.datasus.gov.br/dissemin/publicos/SIM/CID10/DORES/</a> )
SINAN	Sistema de Informação de Agravos de Notificação, monitora doenças e agravos de notificação compulsória.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SINAN/DADOS/FINAIS/">ftp://ftp.datasus.gov.br/dissemin/publicos/SINAN/DADOS/FINAIS/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SINAN/DADOS/FINAIS/">ftp://ftp.datasus.gov.br/dissemin/publicos/SINAN/DADOS/FINAIS/</a> )
SINASC	Sistema de Informações sobre Nascidos Vivos, coleta dados sobre nascimentos no Brasil.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/">ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/">ftp://ftp.datasus.gov.br/dissemin/publicos/SINASC/1996_/Dados/DNRES/</a> )
CNES	Cadastro Nacional de Estabelecimentos de Saúde, registra informações sobre instituições de saúde no Brasil.	<a href="ftp://ftp.datasus.gov.br/dissemin/publicos/CNES/">ftp://ftp.datasus.gov.br/dissemin/publicos/CNES/</a> ( <a href="ftp://ftp.datasus.gov.br/dissemin/publicos/CNES/">ftp://ftp.datasus.gov.br/dissemin/publicos/CNES/</a> )
OpenDataSUS	Portal de dados abertos do SUS, com acesso a diversos conjuntos de dados.	<a href="https://opendatasus.saude.gov.br/">https://opendatasus.saude.gov.br/</a> ( <a href="https://opendatasus.saude.gov.br/">https://opendatasus.saude.gov.br/</a> )

---

# FTP e repositórios públicos do SUS - Outras fontes

Sigla	Descrição	URL
SIDRA	Sistema IBGE de Recuperação Automática, acesso a dados de pesquisas como a PNAD.	<a href="https://sidra.ibge.gov.br/pesquisa/pnadct/tabelas">https://sidra.ibge.gov.br/pesquisa/pnadct/tabelas</a> ( <a href="https://sidra.ibge.gov.br/pesquisa/pnadct/tabelas">https://sidra.ibge.gov.br/pesquisa/pnadct/tabelas</a> )
PNAD	Pesquisa Nacional por Amostra de Domicílios, coleta dados socioeconômicos e demográficos no Brasil.	—
ANVISA	Portal de dados públicos da ANVISA, com informações regulatórias e sanitárias.	<a href="https://dados.anvisa.gov.br/">https://dados.anvisa.gov.br/</a> ( <a href="https://dados.anvisa.gov.br/">https://dados.anvisa.gov.br/</a> ), <a href="https://dados.gov.br/dados/organizacoes/visualizar/agencia-nacional-de-vigilancia-sanitaria-anvisa">https://dados.gov.br/dados/organizacoes/visualizar/agencia-nacional-de-vigilancia-sanitaria-anvisa</a> ( <a href="https://dados.gov.br/dados/organizacoes/visualizar/agencia-nacional-de-vigilancia-sanitaria-anvisa">https://dados.gov.br/dados/organizacoes/visualizar/agencia-nacional-de-vigilancia-sanitaria-anvisa</a> )
SISAB	Sistema de Informação em Saúde para a Atenção Básica (SISAB), Coleta de Dados Simplificado (CDS) e Prontuário Eletrônico do Cidadão (PEC).	<a href="https://sisab.saude.gov.br/">https://sisab.saude.gov.br/</a> ( <a href="https://sisab.saude.gov.br/">https://sisab.saude.gov.br/</a> )

---

Contudo, o **FTP do DATASUS** continua sendo uma das principais fontes de dados brutos utilizados em análises, estudos e políticas públicas na área da saúde.

**Fundamentos práticos dos dados  
ambulatoriais do SUS e  
organização do repositório da  
SABEIS.**



# Intalando bibliotecas

```
if (!requireNamespace("readr", quietly=TRUE)) install.packages("readr")
if (!requireNamespace("dplyr", quietly=TRUE)) install.packages("dplyr")
if (!requireNamespace("stringr", quietly=TRUE)) install.packages("stringr")
if (!requireNamespace("tidyr", quietly=TRUE)) install.packages("tidyr")
# if (!requireNamespace("RCurl", quietly=TRUE)) install.packages("RCurl")
if (!requireNamespace("curl", quietly=TRUE)) install.packages("curl")
if (!requireNamespace("data.table", quietly=TRUE)) install.packages("data.table")
if (!requireNamespace("DT", quietly=TRUE)) install.packages("DT")
library(readr); library(dplyr); library(stringr); library(tidyr)
library(DT); library(data.table); library(shiny); library(curl)
```

- Verifica se o pacote curl está instalado.
- Caso não esteja, instala automaticamente.
- Em seguida, carrega o pacote.

**Operações de ETL de dados tabulados a partir do FTP e organização em repositórios estruturados (RData e SQL).**

# listar csv a partir do FTP

## Código e resultado

```
library(curl)
url <- "sftp://45.231.133.29:7891/srv/ftp/dadosabertos/"
userpwd <- "livre:12345"
h <- new_handle(userpwd = userpwd, dirlistonly = TRUE)
con <- curl(url, handle = h)
arquivos <- readLines(con)
close(con)
head(arquivos)
```

```
## [1] "."
## [2] ".."
## [3] "tf_sia_200801_pa_093_lin_subgrupo304.20251105.134649.csv.gz"
## [4] "tf_sia_200801_pa_004_pro_subgrupo304.20251105.13153497.csv.gz"
## [5] "tf_sia_200801_pa_088_lla_subgrupo304.20251105.756110.csv.gz"
## [6] "tf_sia_200801_pa_045_pom_subgrupo604.20251105.1398.csv.gz"
```

# listar e baixar csv a partir do FTP

## Explicação do código

O código retorna no console um vetor de strings, cada elemento representando um arquivo ou pasta disponível no diretório remoto /srv/ftp/dadosabertos do servidor informado.

- url: endereço do servidor SFTP, incluindo host, porta (7891) e caminho do diretório (/srv/ftp/dadosabertos/).
- userpwd: credenciais no formato "usuário:senha".
- getURL() acessa a URL informada.
- userpwd = userpwd: passa as credenciais de login.
- ftp.use.epsv = FALSE: desabilita *Extended Passive Mode*, evitando problemas de compatibilidade.
- dirlistonly = TRUE: retorna apenas a lista de arquivos/pastas no diretório remoto.
- strsplit() divide o texto retornado em linhas, separando cada arquivo/pasta.
- unlist() transforma a lista resultante em vetor.
- print/head(arquivos) exibe a lista final de arquivos.

**Atividade prática em grupo:  
definição de critérios de seleção  
de usuários do SUS e extração de  
dados para construção de coortes.**

# baixar csv a partir do FTP

## Código e resultado

O script conecta-se ao servidor SFTP, autentica com usuário e senha, baixa o arquivo 0teste\_058\_fen\_subgrupo604.20251104.80541.csv.gz e o salva no computador no diretório de trabalho atual.

```
library(curl)
url <- "sftp://45.231.133.29:7891/srv/ftp/dadosabertos/0teste_058_fen_subgrupo604.20251104.80541.csv.gz"
userpwd <- "livre:12345"
destino <- "0teste_058_fen_subgrupo604.20251104.80541.csv.gz"
h <- new_handle(userpwd = userpwd)
# baixa em binário
curl_download(url, destfile = destino, mode = "wb", handle = h)
cat("Arquivo salvo em:", destino, "\n")

## Arquivo salvo em: 0teste_058_fen_subgrupo604.20251104.80541.csv.gz
```

# Baixando via SFTP/FTP com curl

Este exemplo mostra como baixar um arquivo remoto de um servidor SFTP e salvá-lo localmente no computador usando o pacote curl.

```
library(curl)

url <- "sftp://45.231.133.29:7891/srv/ftp/dadosabertos/0teste_058_fen_subgrupo604.20251104.80541.csv.gz"
userpwd <- "livre:12345"
destino <- "0teste_058_fen_subgrupo604.20251104.80541.csv.gz"

# Cria o handle com credenciais
h <- new_handle(userpwd = userpwd)

# Faz o download binário
curl_download(url, destfile = destino, mode = "wb", handle = h)

# Mensagem de confirmação
cat("Arquivo salvo em:", destino, "\n")
```

- Imprime no console a mensagem confirmando onde o arquivo foi salvo.

# Leitura do arquivo csv.gz baixado

## Código e resultado

```
library(data.table)

arquivo <- "0teste_058_fen_subgrupo604.20251104.80541.csv.gz"
df <- fread(arquivo)

# Ver estrutura
dim(df)  # número de linhas e colunas

## [1] 80541    29

df[1:3,9:11]  # primeiras linhas

##           nu_cnspcn co_procedimento co_cidpri
## 1: 262774798516195      604060025      E700
## 2: 262779456962551      604060025      E700
## 3: 262774788120470      604060025      E700
```



# Leitura do arquivo csv.gz baixado

Use `data.table::fread()` para leitura eficiente de arquivos grandes.

```
if (!requireNamespace("data.table", quietly=TRUE)) install.packages("data.table")
library(data.table)
```

- Carregar em dataframe (data.table)

```
df <- fread(arquivo)
```

`fread()` lê arquivos compactados `.gz` diretamente, sem necessidade de descompactar antes.

Se os arquivos ocuparem muita memória, use o pacote `bigmemory` ou `ff`. Porém, na prática `fread` é otimizado para milhões de linhas.

# Navegue em outras diretrizes

Escolha pela sigla do PCDT:

006\_fer



# dplyr para transformações de dados

## O que é o dplyr

dplyr é um pacote do R usado para **manipular e resumir dados** de forma simples e rápida. Ele segue uma lógica parecida com a do SQL, mas com uma sintaxe mais legível.

# Principais funções

Função	Função prática
<b>filter()</b>	Seleciona linhas com base em condições.
<b>select()</b>	Escolhe colunas específicas.
<b>mutate()</b>	Cria ou altera variáveis.
<b>arrange()</b>	Ordena linhas.
<b>group_by()</b>	Agrupar dados para resumo.
<b>summarise()</b>	Calcula estatísticas (ex.: soma, média, contagem).

---

## Exemplo

```
library(dplyr)
```

```
dados %>%  
  filter(idade > 18) %>%           # mantém apenas adultos  
  mutate(faixa = ifelse(idade > 60, "idoso", "adulto")) %>% # cria nova coluna  
  group_by(faixa) %>%  
  summarise(total = n())          # conta quantos há em cada faixa
```

Esse fluxo lê-se como uma sequência de etapas, conectadas pelo operador %>%, facilitando a leitura: **“Filtre, crie uma nova coluna, agrupe e resuma.”**

## Em resumo

dplyr simplifica a análise de dados no R, tornando o código **mais claro, conciso e eficiente**, mesmo para quem não é programador avançado.

# Enriquecendo atributos - Diretriz

```
library(readr)

td_diretriz_cnv <- read_csv(
  "https://raw.githubusercontent.com/sabeis-ats/bd_geral/refs/heads/main/td_diretriz_cnv.csv"
)

as.data.frame(td_diretriz_cnv)[8:12,5:7]
```

##	sg_pcdt	no_abreviado	no_diretriz_b
## 8	AIJ	artrite_ju	Artrite Idiopática Juvenil (AIJ)
## 9	APS	artrite_ps	Artrite Psoríaca
## 10	ART	artrite_re	Artrite Reativa
## 11	ASM	asma	Asma
## 12	AME	atrofia	Atrofia Muscular Espinhal 5q Tipo 1

# Enriquecendo atributos - Doença

```
library(readr)

td_doenca_cid10 <- read_csv(
  "https://raw.githubusercontent.com/sabeis-ats/bd_geral/refs/heads/main/td_doenca_cid10.csv"
)

as.data.frame(td_doenca_cid10)[8:12,c("co_cid","no_cid","co_capitulo")]
```

##	co_cid	no_cid	co_capitulo
## 8	C171	Jejuno	2
## 9	C481	Partes especificadas do peritonio	2
## 10	C169	Estomago, nao especificado	2
## 11	C155	Terco inferior do esofago	2
## 12	C165	Pequena curvatura do estomago, nao especificada	2

# Enriquecendo atributos - Procedimento

```
library(readr)
```

```
td_sigtap <- read_csv(  
  "https://raw.githubusercontent.com/sabeis-ats/bd_geral/refs/heads/main/td_sigtap.csv"  
)
```

```
as.data.frame(td_sigtap)[8:12,c("co_procedimento","no_procedimento")]
```

```
##      co_procedimento
```

```
## 8      102010480
```

```
## 9      102010498
```

```
## 10     102010013
```

```
## 11     201010011
```

```
## 12     201010119
```

```
##
```

```
## 8  FISCALIZAÇÃO DO USO DE PRODUTOS FUMÍGENOS DERIVADOS DO TABACO EM AMBIENTES COLETIVOS FECHADOS, PÚBLICOS OU
```

```
## 9      LAUDO DE ANÁLISE LABORATORIAL DO PROGRAMA DE MONITORAMENTO DE ALIMENTOS RECEBIDOS PELA VIGILÂNCIA S.
```

```
## 10                                     APLICACAO D
```

```
## 11                                     AMNI
```

```
## 12                                     BIOPSIA D
```



# Enriquecendo atributos - Território

```
library(readr)

td_municipio <- read_csv(
  "https://raw.githubusercontent.com/sabeis-ats/bd_geral/refs/heads/main/municipio_regiao_saude_amazonia_legal_ip
")

as.data.frame(td_municipio)[8:12,c("Codigo_Municipio","Municipio","UF_x")]
```

##	Codigo_Municipio	Municipio	UF_x
## 8	110001	Alta Floresta D'Oeste	Rondonia
## 9	110032	Sao Miguel do Guapore	Rondonia
## 10	110025	Presidente Medici	Rondonia
## 11	110149	Sao Francisco do Guapore	Rondonia
## 12	110005	Cerejeiras	Rondonia

# Cruzando tabelas

```
# csvs: vetor de nomes de arquivos já retornado por listar_csv_py(...)

# 1) tabela com código (3 dígitos) extraído da 5ª parte do nome + qt_registro após o primeiro ponto
codigos_arquivos <- tibble(no_arquivo = arquivos) %>%
  mutate(
    quinta_parte = str_split(no_arquivo, "_", simplify = TRUE)[, 5],
    codigo = str_extract(quinta_parte, "[0-9]{3}"),
    qt_registro = str_extract(no_arquivo, "(?<=\\.)[0-9]+")
  ) %>%
  filter(!is.na(codigo))
```

# Cruzando tabelas

```
# 2) normalizar nu_diretriz para 3 dígitos
if (is.numeric(td_diretriz_cnv$nu_diretriz)) {
  tdc <- td_diretriz_cnv %>%
    mutate(nu_diretriz_chr = sprintf("%03d", as.integer(nu_diretriz)))
} else {
  tdc <- td_diretriz_cnv %>%
    mutate(nu_diretriz_chr = str_pad(str_extract(nu_diretriz, "\\d+"), 3, pad = "0"))
}
```

# Cruzando tabelas

# 3) join e seleção de colunas pedidas

```
pcdt <- codigos_arquivos %>%  
  inner_join(tdc, by = c("codigo" = "nu_diretriz_chr")) %>%  
  mutate(  
    co_cid = str_replace_all(co_cid, ",", " ")  
  ) %>%  
  select(  
    nu_diretriz, no_diretriz, co_cid, sg_pcdt, no_diretriz10, sg_tipo,  
    no_arquivo, qt_registro  
  )
```

```
head(pcdt,3)
```

```
## # A tibble: 3 × 8  
##   nu_diretriz no_diretriz      co_cid sg_pcdt no_diretriz10 sg_tipo no_arquivo  
##   <chr>      <chr>      <chr> <chr>    <chr>      <chr> <chr>  
## 1 093      Linfoma Folicular "C820... LIN      Linfoma Fo    ddt      tf_sia_20...  
## 2 004      Adenocarcinoma de... "C61 ... PRO      A. Próst.    ddt      tf_sia_20...  
## 3 088      Leucemia Linfoblá... "C835... LLA      <NA>        <NA>      tf_sia_20...  
## # i 1 more variable: qt_registro <chr>
```

# Navegue pelos df carregados

td\_municipio

Show2entries

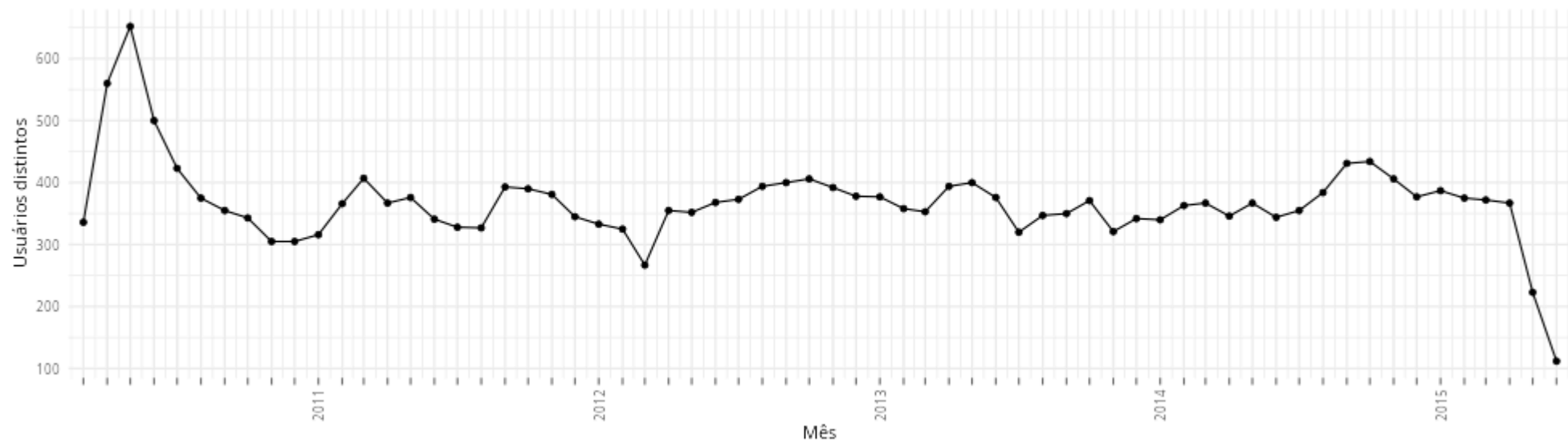
Search:

Codigo_Regiao_do_Pais	Regiao_do_Pais	Codigo_UF	UF_x	Codigo_Macrorregiao_de_Saude	Macrorregiao_de_Saude	Codi
All	All	A		All	All	All
1	Norte	11	Rondonia	1101	MACRORREGIONAL II (CACOAL)	
1	Norte	11	Rondonia	1101	MACRORREGIONAL II (CACOAL)	
Showing 1 to 2 of 5,571 entries						
Previous12345...2,786Next						

# Demanda aferida.

Escolha pela sigla do PCDT:

006\_fer



## Explicação do trecho dplyr — Demanda aferida

O código abaixo resume os dados para calcular o **número de usuários distintos por mês**. Ele reproduz a lógica da query SQL:

```
s <- df %>%  
  filter(!is.na(nu_cnspcn)) %>%           # mantém registros com usuário identificado  
  mutate(  
    mes = ymd(paste0(sprintf("%06d", as.integer(nu_competencia)), "01"))  
  ) %>%                                   # converte YYYYMM → data completa  
  group_by(mes) %>%  
  summarise(  
    usuarios = n_distinct(nu_cnspcn),      # conta usuários únicos no mês  
    .groups = "drop"  
  )
```

```
SELECT  
  nu_competencia,  
  COUNT(DISTINCT nu_cnspcn) AS usuarios  
FROM tf_sia_200801_pa_046_wil_subgrupo604  
GROUP BY 1  
ORDER BY 1;
```

### 1. `filter(!is.na(nu_cnspcn))`

Remove linhas em que o identificador do usuário (`nu_cnspcn`) está ausente, garantindo contagem correta de indivíduos.

### 2. `mutate(mes = ...)`

Cria a variável `mes` em formato de data:

- `sprintf("%06d", ...)` garante que `nu_competencia` tenha sempre 6 dígitos (ex.: 202401);
- concatena "01" para formar uma data completa (YYYYMM01);
- `ymd()` converte para tipo `Date`, permitindo uso no eixo X do gráfico.

### 3. `group_by(mes) + summarise()`

Agrupar os registros por mês e calcular:

- `n_distinct(nu_cnspcn)` → número de usuários distintos.
- `.groups = "drop"` evita manter o agrupamento no *data frame* final.



## Resultado

O *data frame* `s` contém:

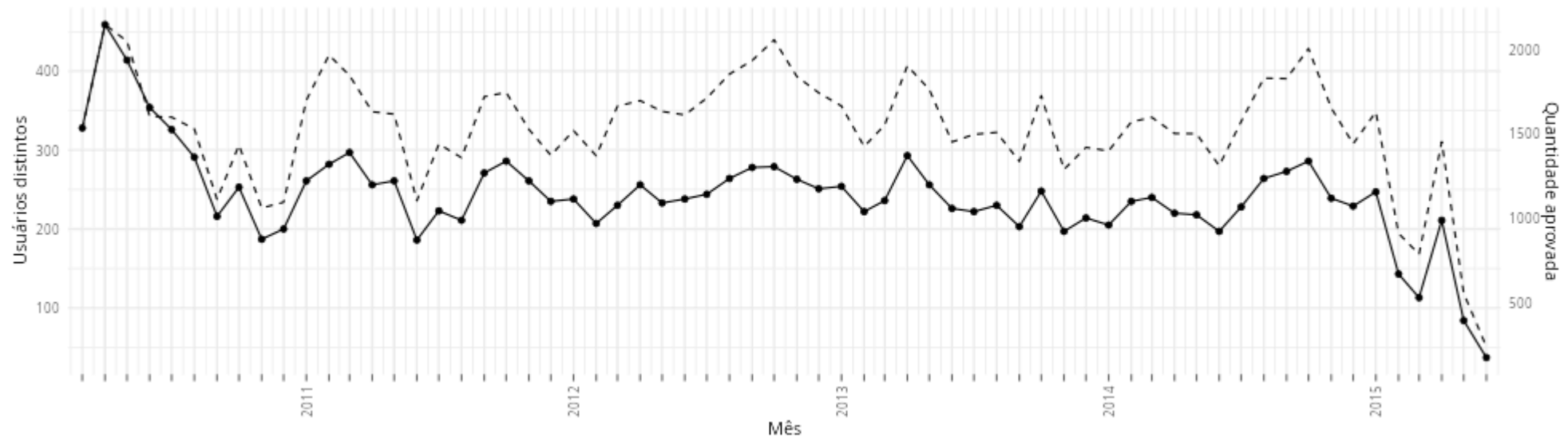
- **mes** → data de referência (primeiro dia do mês de competência);
- **usuarios** → número de usuários distintos naquele mês.

Esse resumo alimenta o gráfico de “Demanda aferida”, que mostra a evolução mensal de usuários atendidos.

# Avaliação pós-incorporação.

Escolha pela sigla do PCDT:

006\_fer



## Explicação do filtro e agregação (dplyr)

O trecho de código abaixo prepara o conjunto de dados antes da visualização, aplicando filtros e agregações que reproduzem a lógica da query SQL original.

```
s <- df %>%  
  # filtro da query: WHERE substr(no_origem,5,2) <> 25  
  filter(!is.na(nu_cnspcn),  
         qt_aprovada > 0 %>%  
  mutate(mes = ymd(paste0(nu_competencia, "01"))) %>%  
  group_by(mes) %>%  
  summarise(  
    usuarios      = n_distinct(nu_cnspcn),  
    qt_aprovada = sum(qt_aprovada, na.rm = TRUE),  
    .groups = "drop"  
  )
```

## 1. Filtro (`filter`)

- `!is.na(nu_cnspcn)` remove registros sem identificação de usuário.
- `qt_aprovada > 0` mantém apenas procedimentos com quantidade aprovada positiva.

## 2. Criação da variável de tempo (`mutate`)

- `mutate(mes = ymd(paste0(nu_competencia, "01")))` transforma o campo `nu_competencia` (formato YYYYMM) em data (YYYY-MM-01) para uso correto no eixo X do gráfico.

## 3. Agregação (`group_by + summarise`)

- `group_by(mes)` agrupa por mês de competência.
- `n_distinct(nu_cnspcn)` calcula o número de usuários distintos no mês.
- `sum(qt_aprovada, na.rm = TRUE)` soma a quantidade aprovada, ignorando NA.
- `.groups = "drop"` evita manter a estrutura de agrupamento após a sumarização.

O resultado é um *data frame* resumido por mês, com duas métricas principais:

- **usuarios**: número de usuários únicos.
- **qt\_aprovada**: total de procedimentos aprovados.

**Informações  
complementares**

# Referências complementares

- Ferré, Felipe. Infoestrutura para apoio à decisão estratégica no SUS. In: Alethele de Oliveira Santo; Luciana Tolêdo Lopes. (Org.). Coleção Covid-19. 1ed.Brasília: Conselho Nacional de Secretários de Saúde, 2021, v. 6, p. 114-135. [link](https://bit.ly/35D0k0j) (<https://bit.ly/35D0k0j>)
- Prova de conceito SABEIS de 2019 [painel](https://labxss.shinyapps.io/sabeis_pcdt3/) ([https://labxss.shinyapps.io/sabeis\\_pcdt3/](https://labxss.shinyapps.io/sabeis_pcdt3/))
- Prova de conceito SABEIS de 2019 [wiki](https://bookdown.org/labxss/sabeis_pcdt5/) ([https://bookdown.org/labxss/sabeis\\_pcdt5/](https://bookdown.org/labxss/sabeis_pcdt5/))
- Prova de conceito SABEIS de 2019 [git](https://github.com/labxss/sabeis_pcdt) ([https://github.com/labxss/sabeis\\_pcdt](https://github.com/labxss/sabeis_pcdt))