

Guia

Pacote RHCgen

Sumário

Introdução	3
Instalação	4
Principais Funções	5
Função <i>lerarquivoDBF()</i>	5
Função <i>renomear_colunas()</i>	8
Função <i>modificar_tipo_variavel()</i>	10
Função <i>recodificar_variaveis()</i>	13
Função <i>renomear_siglas_estados()</i>	16
Função <i>renomear_codigo_municipio_residencia()</i>	19
Função <i>renomear_codigo_municipio_hospital()</i>	21
Função <i>renomear_CID_3digitos()</i>	24
Função <i>renomear_CID_4digitos()</i>	27
Função <i>renomear_CNES()</i>	30
Função <i>renomear_tipo_histologico()</i>	33
Função <i>renomear_clinica()</i>	36
Função <i>renomear_estadiamento_clinico()</i>	39
Função <i>ordenando_colunas()</i>	42
Função <i>analise_qualidade_dos_dados()</i>	44
Função <i>analise_completude()</i>	45
Função <i>analise_completude_ano()</i>	48
Função <i>tabela_inconsistencias()</i>	51
Função <i>construir_banco()</i>	56
Função <i>filtrar_banco()</i>	59

Introdução

O RHCgen é um pacote desenvolvido em R para automatizar e padronizar o processamento de dados do Registro Hospitalar de Câncer (RHC) no Brasil. Sua principal função é facilitar a manipulação, organização e análise de grandes volumes de dados do IntegradorRHC (IRHC), reduzindo a necessidade de processos manuais e aumentando a eficiência na preparação dos dados para estudos epidemiológicos e clínicos.

O RHC armazena informações detalhadas sobre os casos de câncer atendidos em hospitais habilitados pelo Sistema Único de Saúde (SUS), abrangendo aspectos clínicos, demográficos e terapêuticos. No entanto, os dados disponibilizados pelo Instituto Nacional de Câncer (INCA) exigem diversas etapas de pré-processamento antes de serem utilizados em análises estatísticas. O RHCgen foi criado para suprir essa necessidade, permitindo a aplicação de funções automatizadas que garantem maior consistência, reprodutibilidade e qualidade na organização dos dados.

O pacote possibilita:

- Leitura e combinação de arquivos do RHC em formato DBF.
- Conversão e padronização de variáveis (ex.: nomes de colunas, códigos de estadiamento, CID-O, CNES).
- Verificação e correção de inconsistências, como idades inválidas e erros cronológicos nas datas.
- Análises de qualidade dos dados, incluindo a avaliação de completude e classificação conforme critérios de Romero & Cunha (2006).
- Filtragem personalizada dos dados, facilitando recortes específicos por tipo de câncer, período de diagnóstico, unidade hospitalar, entre outros.

A utilização do RHCgen reduz significativamente o tempo de preparação dos dados e minimiza erros associados ao manuseio manual das bases do RHC, permitindo que pesquisadores e profissionais de saúde se concentrem diretamente na análise epidemiológica e clínica.

Instalação

O RHCgen está disponível no GitHub e pode ser instalado diretamente no R. Para garantir uma instalação sem problemas, siga as etapas abaixo:

1. Instale o pacote *remotes*

O *remotes* permite instalar pacotes diretamente do GitHub. Se ainda não estiver instalado, execute:

```
install.packages("remotes")
```

2. Instale o RHCgen a partir do GitHub

Agora, instale o pacote executando:

```
remotes::install_github("andersonlineu/RHCgen")
```

3. Carregue o pacote no R

Após a instalação, carregue o pacote normalmente:

```
library(RHCgen)
```

Dependências

O RHCgen requer alguns pacotes adicionais para funcionar corretamente. Um deles é o *foreign*, responsável por ler arquivos DBF. Esse pacote é instalado automaticamente ao utilizar o RHCgen pela primeira vez.

Caso queira instalar manualmente, utilize:

```
install.packages("foreign")
```

Requisitos do sistema

- **R versão 4.3.3** ou superior.
- **RStudio** recomendado para um melhor ambiente de desenvolvimento.
- Conexão com a internet para baixar e instalar pacotes do GitHub.

Principais Funções

Função *lerarquivoDBF()*

Descrição

A função *lerarquivoDBF()* é utilizada para ler e combinar múltiplos arquivos DBF provenientes do Integrador RHC. Além disso, a função adiciona uma nova coluna chamada *Ano_do_Banco*, que é extraída do nome do arquivo, facilitando análises temporais.

Parâmetros

A função não requer parâmetros externos, pois opera diretamente nos arquivos DBF presentes no diretório de trabalho.

Retorno

- Retorna um dataframe contendo a combinação de todos os arquivos DBF disponíveis no diretório de trabalho.
- Se não houver arquivos disponíveis ou se ocorrerem erros na leitura, a função retorna NULL.

Pré-requisitos

- Os arquivos DBF devem seguir o padrão de nomenclatura do Integrador RHC, iniciando com "rhc" seguido pelo ano (rhc20.dbf para 2020, por exemplo).
- Todos os arquivos precisam estar na pasta de trabalho atual (`getwd()`).

Funcionamento

A função executa os seguintes passos:

1. Carrega o pacote *foreign* (instala se necessário).
2. Identifica arquivos DBF no diretório de trabalho que comecem com "rhc" (exemplo: rhc20.dbf).
3. Lê os arquivos DBF usando *foreign::read.dbf()*.
4. Extrai o ano do nome do arquivo e adiciona uma nova coluna chamada *Ano_do_Banco*.
5. Garante que todas as colunas sejam padronizadas, adicionando colunas ausentes com NA para evitar problemas ao combinar os dataframes.

6. Combina todos os dataframes em um único objeto.
7. Retorna o dataframe final contendo os dados de todos os arquivos lidos.

Mensagens Importantes

- Se nenhum arquivo for encontrado → Aparece um alerta vermelho informando que não há arquivos "rhc*.dbf" no diretório.
- Se um arquivo estiver vazio → Aparece um alerta amarelo informando que ele não contém dados.
- Se ocorrer erro na leitura de algum arquivo → A função imprime a mensagem de erro correspondente.
- Se a estrutura dos arquivos for inconsistente (diferente número de colunas) → A função tenta padronizar as colunas, mas pode exibir erro caso as colunas não sejam compatíveis.

Exemplo de Uso

Suponha que você tenha os seguintes arquivos na pasta de trabalho:

```
rhc20.dbf
rhc21.dbf
rhc22.dbf
```

Você pode executar a função da seguinte forma:

```
# Executando a função
dados_RHC_combinados <- lerarquivoDBF()
```

Se a leitura for bem-sucedida:

- dados_RHC_combinados conterá os dados combinados dos arquivos DBF, com uma coluna adicional chamada Ano_do_Banco.
- Se houver erro ou nenhum arquivo for encontrado, a função retornará NULL.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
------	----------------	---------

> Nenhum arquivo 'rhc*.dbf' encontrado.	Não há arquivos DBF no diretório de trabalho	Verifique se os arquivos estão na pasta correta e renomeados corretamente
Erro ao ler o arquivo:	Arquivo corrompido ou formato inválido	Tente abrir o arquivo manualmente e verificar sua integridade
Erro: Os dataframes não têm o mesmo número de colunas.	Diferenças na estrutura das tabelas entre os arquivos DBF	Verifique se todos os arquivos seguem o mesmo layout de colunas

A função *lerarquivoDBF()* simplifica a importação e fusão de arquivos DBF do RHC, permitindo uma análise eficiente de dados ao longo dos anos. Ao garantir que todos os arquivos sigam um padrão, ela otimiza o processamento e reduz possíveis erros.

Função *renomear_colunas()*

Descrição

A função `renomear_colunas` tem como objetivo renomear as colunas de um dataframe do Registro Hospitalar de Câncer (RHC), substituindo nomes técnicos por descrições mais intuitivas. Isso facilita a leitura e a interpretação dos dados.

Parâmetros

- Um dataframe contendo as colunas a serem renomeadas.

Retorno

- Retorna um dataframe modificado, onde as colunas encontradas no mapeamento predefinido foram substituídas pelos nomes descritivos.
- Se alguma coluna do mapeamento não estiver presente no dataframe, ela será ignorada, e uma mensagem será exibida.

Funcionamento

1. Define um dicionário de mapeamento entre os nomes originais das colunas e seus respectivos nomes descritivos.
2. Verifica a existência das colunas no dataframe e, se encontradas, renomeia-as.
3. Exibe mensagens informativas, indicando quais colunas foram renomeadas e quais não foram encontradas.
4. Retorna o dataframe atualizado com os novos nomes de colunas.

Mensagens Importantes

- Se uma **coluna for encontrada e renomeada**, a função exibe:

```
Renomeando coluna: TPCASO para: Tipo_de_Caso
```

- Se uma **coluna não for encontrada**, a função exibe:

```
A coluna: ANOPRIDI não existe no dataframe e será ignorada.
```

- Ao final, a função exibe uma mensagem de sucesso com o total de colunas renomeadas.

Exemplo de Uso

Suponha que o dataframe `dados_RHC` tenha as seguintes colunas:

```
colnames(dados_RHC)
# [1] "TPCASO" "SEXO" "IDADE" "HISTFAMC" "ANOPRIDI" "DTINITRT"
```

Para renomear as colunas, execute:

```
dados_RHC_renomeado <- renomear_colunas(dados_RHC)
```

Saída esperada:

```
colnames(dados_RHC_renomeado)
# [1] "Tipo_de_Caso" "Sexo" "Idade" "Historico_Familiar_Cancer" "Ano_Primeiro_Diagnostico"
"Ano_Inicio_Tratamento"
```

Possíveis Problemas e Soluções

Problema	Causa Possível	Solução
Nenhuma coluna foi renomeada	As colunas do dataframe não correspondem às do mapeamento	Verifique se os nomes das colunas no dataframe seguem exatamente os nomes usados na função
Mensagens informando que algumas colunas não existem	O dataframe não contém algumas colunas do mapeamento	Isso é esperado. Se necessário, revise a estrutura do dataframe

A função **renomear_colunas()** simplifica a manipulação de dados do **RHC**, tornando as variáveis mais fáceis de interpretar. Essa função é útil para padronizar os dados antes de realizar análises estatísticas ou visualizações.

Função *modificar_tipo_variavel()*

Descrição

A função *modificar_tipo_variavel()* ajusta os tipos de variáveis de um dataframe do Registro Hospitalar de Câncer (RHC), garantindo que cada variável tenha um formato adequado para análises.

- Todas as variáveis são convertidas para character, exceto:
 - Colunas de data, que são transformadas de character para Date (YYYY-MM-DD).
 - A variável Idade, que é convertida para numérico.

Parâmetros

- Um dataframe contendo as variáveis a serem convertidas.

Retorno

- Retorna um dataframe com os tipos de variáveis modificados:
 - Variáveis não listadas como data ou idade → convertidas para character.
 - Variáveis de data (Data_Diagnostico, Data_Triagem, Data_Primeira_Consulta, Data_Inicio_Primeiro_Tratamento, Data_Obito) → convertidas para Date.
 - Variável Idade → convertida para numérico.

Funcionamento

1. Converte todas as variáveis para character, exceto Idade.
2. Converte Idade para numérico.
3. Converte variáveis de data para Date (YYYY-MM-DD), emitindo um alerta caso a conversão resulte em NA (o que indica erro no formato da data).

Mensagens Importantes

- Se uma variável for convertida para character, a função imprime:

```
Convertendo a variável: Sexo para character.
```

- Se a coluna **Idade** for convertida para numérico, a função imprime:

```
Convertendo a variável 'Idade' para numérico.
```

- Se uma variável de data for convertida com sucesso, a função imprime:

```
Convertendo a variável: Data_Diagnostico para data.
```

- Se uma conversão de data resultar em NA, a função alerta:

```
Aviso: A conversão da variável Data_Obito resultou em NAs. Verifique o formato da data.
```

- Mensagem final ao concluir a conversão:

```
> Conversão de tipos de variáveis concluída com sucesso.
```

Exemplo de Uso

Exemplo 1: Dataframe original

```
dados_RHC <- data.frame(
  Sexo = c("M", "F"),
  Idade = c("045", "060"),
  Data_Diagnostico = c("15/06/2021", "20/07/2022"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_modificado <- modificar_tipo_variavel(dados_RHC)
```

Saída esperada

```
str(dados_RHC_modificado)
# 'data.frame': 2 obs. de 3 variáveis:
# $ Sexo      : chr "M" "F"
# $ Idade      : num 45 60
# $ Data_Diagnostico: Date, format: "2021-06-15" "2022-07-20"
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
------	----------------	---------

Aviso: A conversão da variável Data_Diagnostico resultou em NAs.	O formato da data está incorreto (exemplo: "20210615")	Verifique se as datas estão no formato "dd/mm/yyyy" antes da conversão
Error in as.numeric(data\$Idade) : invalid 'NA' argument	Alguma entrada na coluna Idade contém texto não numérico	Use summary(dados_RHC\$Idade) para verificar valores inválidos antes da conversão

A função *modificar_tipo_variavel()* é útil para padronizar os tipos de variáveis do RHC, preparando os dados para análise sem necessidade de ajustes manuais. Isso melhora a compatibilidade com modelos estatísticos e facilita a manipulação dos dados.

Função *recodificar_variaveis()*

Descrição

A função *recodificar_variaveis()* transforma variáveis categóricas codificadas numericamente em fatores com níveis descritivos. O objetivo é melhorar a legibilidade dos dados do Registro Hospitalar de Câncer (RHC) e garantir consistência na análise.

Parâmetros

- Um dataframe contendo as variáveis categóricas a serem recodificadas.

Retorno

- Retorna um dataframe com as variáveis recodificadas em fatores com níveis descritivos.
- Se uma variável esperada não estiver presente, a função exibe um aviso e continua o processo com as variáveis disponíveis.

Funcionamento

1. Verifica quais variáveis esperadas estão presentes no dataframe.
2. Para cada variável encontrada, substitui os valores numéricos pelos rótulos correspondentes.
3. Converte a variável para o tipo factor, garantindo que os níveis sejam coerentes.
4. Exibe mensagens indicando o progresso, incluindo:
 - Variáveis recodificadas.
 - Variáveis ausentes no dataframe.
 - Mensagem final com o número total de variáveis recodificadas.

Mensagens Importantes

- Se **uma variável for recodificada**, a função imprime:

```
Recodificando 'Sexo'.
```

- Se **uma variável estiver ausente**, a função exibe:

```
As seguintes variáveis estão ausentes no dataframe: Consumo_Alcool, Tabagismo
```

- Ao final do processo, a função exibe:

```
> Recodificação das variáveis concluída. Total: 10 com sucesso
```

Exemplo de Uso

Exemplo 1: Dataframe original

```
dados_RHC <- data.frame(
  Tipo_de_Caso = c("1", "2"),
  Sexo = c("1", "2"),
  Raca_Cor = c("1", "4"),
  Escolaridade = c("2", "5"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_recodificado <- recodificar_variaveis(dados_RHC)
```

Saída esperada

```
str(dados_RHC_recodificado)
# 'data.frame': 2 obs. de 4 variáveis:
# $ Tipo_de_Caso : Factor w/ 3 levels "Analítico","Não Analítico","Outro": 1 2
# $ Sexo          : Factor w/ 3 levels "Masculino","Feminino","Outro": 1 2
# $ Raca_Cor      : Factor w/ 6 levels "Branca","Preta",...: 1 4
# $ Escolaridade  : Factor w/ 7 levels "Nenhuma","Fundamental incompleto",...: 2 5
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
Algumas variáveis não foram recodificadas	Elas não existem no dataframe	Verifique a estrutura do dataframe com <code>names(dados_RHC)</code> antes de executar a função
Valores não mapeados continuam como números	O dataframe contém valores inesperados	Certifique-se de que os códigos estão corretos antes de rodar a função

A função *recodificar_variaveis()* simplifica a interpretação dos dados do RHC, convertendo códigos numéricos em rótulos descritivos. Isso facilita análises estatísticas, relatórios e visualizações.

Função *renomear_siglas_estados()*

Descrição

A função `renomear_siglas_estados` converte as siglas dos estados brasileiros nas variáveis `Estado_Residencia` e `UF_Unidade_Hospital` em seus respectivos nomes completos. As informações são adicionadas em novas colunas chamadas:

- `Nome_Estado_Residencia` → Nome completo do estado de residência do paciente.
- `Nome_Estado_Hospital` → Nome completo do estado da unidade hospitalar.

Parâmetros

- Um dataframe contendo as colunas `Estado_Residencia` e/ou `UF_Unidade_Hospital`.

Retorno

- Retorna um dataframe atualizado, contendo as colunas originais e as novas colunas com os nomes completos dos estados.
- Se as colunas necessárias não existirem, a função emite um alerta e interrompe a execução.

Funcionamento

1. Verifica se as colunas `Estado_Residencia` e `UF_Unidade_Hospital` estão presentes no dataframe.
2. Usa um dicionário (`state_map`) para mapear as siglas para seus respectivos nomes completos.
3. Cria novas colunas com os nomes completos dos estados, preservando as siglas originais.
4. Exibe mensagens indicando o progresso, incluindo:
 - Confirmação de colunas adicionadas.
 - Avisos caso alguma coluna esteja ausente.
 - Erro crítico se nenhuma coluna for encontrada.

Mensagens Importantes

- Se uma **coluna for encontrada e processada**, a função exibe:

Nomes completos dos estados para 'Estado_Residencia' adicionados.

- Se uma **coluna não for encontrada**, a função alerta:

A coluna 'UF_Unidade_Hospital' não foi encontrada no dataframe.

- Se **nenhuma das colunas esperadas estiver presente**, a função interrompe a execução com:

Nenhuma das colunas necessárias ('Estado_Residencia' ou 'UF_Unidade_Hospital') foi encontrada no dataframe. Função interrompida.

- **Mensagem final ao concluir a conversão:**

> Mapeamento das siglas dos estados concluído. Foram adicionadas 2 coluna(s) no dataframe.

Exemplo de Uso

Exemplo 1: Dataframe original

```
dados_RHC <- data.frame(
  Estado_Residencia = c("SP", "RJ", "MG"),
  UF_Unidade_Hospital = c("PR", "BA", "DF"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_siglas_estados(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Estado_Residencia UF_Unidade_Hospital Nome_Estado_Residencia Nome_Estado_Hospital
# 1          SP          PR          São Paulo          Paraná
# 2          RJ          BA          Rio de Janeiro          Bahia
# 3          MG          DF          Minas Gerais          Distrito Federal
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
------	----------------	---------

A coluna 'Estado_Residencia' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada	Verifique se a coluna está presente com <code>names(dados_RHC)</code>
Nenhuma das colunas necessárias foi encontrada. Função interrompida.	O dataframe não tem Estado_Residencia nem UF_Unidade_Hospital	Certifique-se de que as colunas existem antes de executar a função

A função *renomear_siglas_estados()* facilita a leitura e interpretação dos dados do RHC, garantindo que os estados brasileiros sejam representados por seus nomes completos. Isso melhora a qualidade dos relatórios e análises.

Função *renomear_codigo_municipio_residencia()*

Descrição

A função *renomear_codigo_municipio_residencia()* converte os códigos dos municípios brasileiros na coluna `Codigo_Municipio_Residencia` em seus respectivos nomes completos. A informação é adicionada em uma nova coluna chamada:

- `Nome_Municipio_Residencia` → Nome completo do município de residência do paciente.

Parâmetros

- Um dataframe contendo a coluna `Codigo_Municipio_Residencia`.

Retorno

- Retorna um dataframe atualizado, contendo as colunas originais e a nova coluna com os nomes completos dos municípios.
- Se a coluna `Codigo_Municipio_Residencia` não existir, a função interrompe a execução e exibe um erro.

Funcionamento

1. Verifica se a coluna `Codigo_Municipio_Residencia` está presente no dataframe.
2. Usa um dicionário (map) para mapear os códigos de municípios para seus respectivos nomes completos.
3. Cria a nova coluna `Nome_Municipio_Residencia` com os nomes completos dos municípios, preservando os códigos originais.
4. Exibe mensagens indicando o progresso, incluindo:
 - Confirmação da conversão.
 - Erro crítico caso a coluna esperada não esteja presente.

Mensagens Importantes

- Se a **coluna for encontrada e processada**, a função exibe:

Nomes completos dos municípios adicionados.

- Se a **coluna não for encontrada**, a função interrompe a execução com:

A coluna 'Codigo_Municipio_Residencia' não foi encontrada no dataframe. Função interrompida.

- **Mensagem final ao concluir a conversão:**

> Ajuste dos códigos dos municípios concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada Nome_Municipio_Residencia.

Exemplo de Uso

Exemplo 1: Dataframe original

```
dados_RHC <- data.frame(
  Codigo_Municipio_Residencia = c("1100015", "1200013", "1300029"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_codigo_municipio_residencia(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Codigo_Municipio_Residencia Nome_Municipio_Residencia
# 1          1100015      Alta Floresta D'Oeste
# 2          1200013      Acrelândia
# 3          1300029      Alvarães
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Codigo_Municipio_Residencia' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada	Verifique se a coluna está presente com <code>names(dados_RHC)</code>

A função `renomear_codigo_municipio_residencia()` facilita a leitura e interpretação dos dados do RHC, garantindo que os municípios brasileiros sejam representados por seus nomes completos. Isso melhora a qualidade dos relatórios e análises.

Função *renomear_codigo_municipio_hospital()*

Descrição

A função *renomear_codigo_municipio_hospital()* converte os códigos dos municípios brasileiros na coluna `Municipio_Unidade_Hospital` em seus respectivos nomes completos. A informação é adicionada em uma nova coluna chamada:

- `Nome_Municipio_Hospital` → Nome completo do município onde está localizada a unidade hospitalar.

Parâmetros

- Um dataframe contendo a coluna `Municipio_Unidade_Hospital`.

Retorno

Retorna um dataframe atualizado, contendo as colunas originais e a nova coluna com os nomes completos dos municípios.

- Se a coluna `Municipio_Unidade_Hospital` não existir, a função interrompe a execução e exibe um erro.

Funcionamento

1. Verifica se a coluna `Municipio_Unidade_Hospital` está presente no dataframe.
2. Usa um dicionário (map) para mapear os códigos de municípios para seus respectivos nomes completos.
3. Cria a nova coluna `Nome_Municipio_Hospital` com os nomes completos dos municípios, preservando os códigos originais.
4. Exibe mensagens indicando o progresso, incluindo:
 - Confirmação da conversão.
 - Erro crítico caso a coluna esperada não esteja presente.

Mensagens Importantes

- Se a coluna for encontrada e processada, a função exibe:

Nomes completos dos municípios adicionados.

- Se a coluna não for encontrada, a função interrompe a execução com:

A coluna 'Municipio_Unidade_Hospital' não foi encontrada no dataframe. Função interrompida.

- Mensagem final ao concluir a conversão:

> Ajuste dos códigos dos municípios concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada Nome_Municipio_Hospital.

Exemplo de Uso

Exemplo 1: Dataframe original

```
# Criando um dataframe de exemplo
dados_RHC <- data.frame(
  Municipio_Unidade_Hospital = c("1100015", "1200013", "1300029"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
# Convertendo os códigos para nomes
dados_RHC_convertido <- renomear_codigo_municipio_hospital(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Municipio_Unidade_Hospital Nome_Municipio_Hospital
# 1          1100015      Alta Floresta D'Oeste
# 2          1200013          Acrelândia
# 3          1300029          Alvarães
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Municipio_Unidade_Hospital' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada	Verifique se a coluna está presente com <code>names(dados_RHC)</code>

A função *renomear_codigo_municipio_hospital()* facilita a leitura e interpretação dos dados do RHC, garantindo que os municípios brasileiros sejam representados por seus nomes completos. Isso melhora a qualidade dos relatórios e análises.

Função *renomear_CID_3digitos()*

Descrição

A função *renomear_CID_3digitos()* converte os códigos CID-O de 3 dígitos presentes na coluna *Localizacao_Primary_3D* para seus respectivos nomes completos. A informação é adicionada em uma nova coluna denominada:

- **CID3d** → Nome completo do CID-O de 3 dígitos correspondente à localização primária do tumor.

Parâmetros

- Um dataframe contendo a coluna *Localizacao_Primary_3D*.

Retorno

A função retorna um dataframe atualizado, mantendo as colunas originais e adicionando a nova coluna *CID3d* com os nomes completos das classificações CID-O de 3 dígitos.

Caso a coluna *Localizacao_Primary_3D* não exista no dataframe, a execução é interrompida e um erro é exibido.

Funcionamento

1. Verifica se a coluna *Localizacao_Primary_3D* está presente no dataframe.
2. Utiliza um mapeamento (dicionário nomeado) para associar os códigos CID-O de 3 dígitos a seus respectivos nomes completos.
3. Cria a nova coluna *CID3d*, preservando os códigos originais e adicionando a descrição correspondente.
4. Chama a função *classificacao_CID_0()* para aplicar a classificação das neoplasias malignas.
5. Exibe mensagens informando o progresso do processo, incluindo:
 - Confirmação da conversão bem-sucedida.
 - Erro crítico caso a coluna esperada não esteja presente no dataframe.

Mensagens Importantes

- Se a coluna `Localizacao_Primary_3D` for encontrada e processada corretamente, a função exibe:

Nomes completos dos CID-O de 3 dígitos adicionados.

- Se a coluna `Localizacao_Primary_3D` não for encontrada, a execução é interrompida e um erro é exibido:

A coluna 'Localizacao_Primary_3D' não foi encontrada no dataframe. Função interrompida.

- Ao concluir a conversão com sucesso, a função exibe:

> Ajuste dos códigos CID de 3 dígitos concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada CID3d.

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Localizacao_Primary_3D = c("C00", "C16", "C50"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_CID_3digitos(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Localizacao_Primary_3D      CID3d
# 1          C00      Neoplasia maligna do lábio
# 2          C16      Neoplasia maligna do estômago
# 3          C50      Neoplasia maligna da mama
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
------	----------------	---------

A coluna 'Localizacao_Primary_3D' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .
--	---	---

A função *renomear_CID_3digitos()* permite a conversão dos códigos CID-O de 3 dígitos para seus respectivos nomes completos, facilitando a interpretação dos dados oncológicos. Isso melhora a padronização e a clareza nas análises do RHC.

Função *renomear_CID_4digitos()*

Descrição

A função `renomear_CID_4digitos` converte os códigos CID-O de 4 dígitos presentes na coluna `Localizacao_Primary_4D` para seus respectivos nomes completos. A informação é adicionada em uma nova coluna denominada:

- **CID4d** → Nome completo do CID-O de 4 dígitos correspondente à localização primária do tumor.

Parâmetros

- Um dataframe contendo a coluna `Localizacao_Primary_4D`.

Retorno

A função retorna um dataframe atualizado, mantendo as colunas originais e adicionando a nova coluna `CID4d` com os nomes completos das classificações CID-O de 4 dígitos.

Caso a coluna `Localizacao_Primary_4D` não exista no dataframe, a execução é interrompida e um erro é exibido.

Funcionamento

1. Verifica se a coluna `Localizacao_Primary_4D` está presente no dataframe.
2. Utiliza um mapeamento (dicionário nomeado) para associar os códigos CID-O de 4 dígitos a seus respectivos nomes completos.
3. Cria a nova coluna `CID4d`, preservando os códigos originais e adicionando a descrição correspondente.
4. Exibe mensagens informando o progresso do processo, incluindo:
 - Confirmação da conversão bem-sucedida.
 - Erro crítico caso a coluna esperada não esteja presente no dataframe.

Mensagens Importantes

- Se a coluna `Localizacao_Primary_4D` for encontrada e processada corretamente, a função exibe:

Nomes completos dos CID-O de 4 dígitos adicionados.

- Se a coluna `Localizacao_Primary_4D` não for encontrada, a execução é interrompida e um erro é exibido:

A coluna 'Localizacao_Primary_4D' não foi encontrada no dataframe. Função interrompida.

- Ao concluir a conversão com sucesso, a função exibe:

> Ajuste dos códigos CID-O de 4 dígitos concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada CID4d.

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Localizacao_Primary_4D = c("C00.0", "C16.1", "C50.9"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_CID_4digitos(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Localizacao_Primary_4D          CID4d
# 1          C00.0      Neoplasia maligna do lábio superior externo
# 2          C16.1      Neoplasia maligna do fundo do estômago
# 3          C50.9      Neoplasia maligna da mama, não especificada
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Localizacao_Primary_4D' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .

A função *renomear_CID_4digitos()* permite a conversão dos códigos CID-O de 4 dígitos para seus respectivos nomes completos, facilitando a interpretação dos dados oncológicos. Isso melhora a padronização e a clareza nas análises do RHC.

Função *renomear_CNES()*

Descrição

A função *renomear_CNES()* converte os códigos CNES presentes na coluna *CNES_Hospital* para os nomes completos dos respectivos estabelecimentos de saúde. A informação é adicionada em uma nova coluna denominada:

- *Estabelecimento_Hospitalar* → Nome completo do estabelecimento de saúde correspondente ao código CNES.

Parâmetros

- Um dataframe contendo a coluna *CNES_Hospital*.

Retorno

A função retorna um dataframe atualizado, mantendo as colunas originais e adicionando a nova coluna *Estabelecimento_Hospitalar* com os nomes completos dos estabelecimentos hospitalares.

Caso a coluna *CNES_Hospital* não exista no dataframe, a execução é interrompida e um erro é exibido.

Funcionamento

1. Verifica se a coluna *CNES_Hospital* está presente no dataframe.
2. Utiliza um mapeamento (dicionário nomeado) para associar os códigos CNES aos seus respectivos nomes completos.
3. Cria a nova coluna *Estabelecimento_Hospitalar*, preservando os códigos originais e adicionando a descrição correspondente.
4. Exibe mensagens informando o progresso do processo, incluindo:
 - Confirmação da conversão bem-sucedida.
 - Erro crítico caso a coluna esperada não esteja presente no dataframe.

Mensagens Importantes

- Se a coluna *CNES_Hospital* for encontrada e processada corretamente, a função exibe:

Nomes completos dos estabelecimentos adicionados.

- Se a coluna CNES_Hospital não for encontrada, a execução é interrompida e um erro é exibido:

A coluna 'CNES_Hospital' não foi encontrada no dataframe. Função interrompida.

- Ao concluir a conversão com sucesso, a função exibe:

> Ajuste dos códigos CNES concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada Estabelecimento_Hospitalar.

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  CNES_Hospital = c("2001586", "2005417", "2007037"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_CNES(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
#   CNES_Hospital      Estabelecimento_Hospitalar
# 1    2001586      Hospital da Fundação Hospitalar Estadual do Acre
# 2    2005417      Complexo Hospitalar Manoel André - CHAMA
# 3    2007037      Hospital da Santa Casa de Misericórdia de Maceió
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'CNES_Hospital' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .

A função *renomear_CNES()* permite a conversão dos códigos CNES para os nomes completos dos estabelecimentos de saúde, facilitando a interpretação dos dados hospitalares no contexto do RHC. Isso melhora a padronização e a clareza nas análises.

Função *renomear_tipo_histologico()*

Descrição

Esta função mapeia os códigos de Tipo Histológico da coluna `Tipo_Histologico` para os nomes completos, adicionando a informação em uma nova coluna:

- `Tipo_Histologico_Completo` → Nome completo do tipo histológico correspondente ao código.

Parâmetros

- Um dataframe contendo a coluna `Tipo_Histologico`.

Retorno

A função retorna o dataframe original com a nova coluna `Tipo_Histologico_Completo`, mantendo todas as colunas originais e adicionando a descrição correspondente.

Caso a coluna `Tipo_Histologico` não exista, a execução é interrompida e um erro é exibido.

Funcionamento

1. Verifica se a coluna `Tipo_Histologico` está presente no dataframe.
2. Utiliza um mapeamento (dicionário nomeado) para associar os códigos de tipo histológico aos seus respectivos nomes completos.
3. Cria a nova coluna `Tipo_Histologico_Completo`, preservando os códigos originais.
4. Exibe mensagens de progresso e finalização do processo.
 - Se a coluna não for encontrada, a função exibe um erro crítico e interrompe a execução.

Mensagens Importantes

- Se a coluna `Tipo_Histologico` for encontrada e processada corretamente, a função exibe:

Nomes completos dos Tipos Histológicos adicionados.

- Se a coluna `Tipo_Histologico` não for encontrada, a execução é interrompida e um erro é exibido:

A coluna 'Tipo_Histologico' não foi encontrada no dataframe. Função interrompida.

- Ao concluir a conversão com sucesso, a função exibe:

```
> Ajuste dos códigos de Tipo Histológico concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada Tipo_Histologico_Completo.
```

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Tipo_Histologico = c("8000/3", "8010/3", "8140/3"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_tipo_histologico(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Tipo_Histologico      Tipo_Histologico_Completo
# 1   8000/3          Neoplasia maligna
# 2   8010/3          Carcinoma SOE
# 3   8140/3          Adenocarcinoma SOE
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Tipo_Histologico' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .

A função `renomear_tipo_histologico()` permite a conversão de códigos de Tipo Histológico para nomes completos, facilitando a análise e interpretação dos dados oncológicos no contexto do RHC. Isso melhora a legibilidade e padronização dos registros.

Função *renomear_clinica()*

Descrição

Essa função mapeia os códigos das clínicas registrados nas colunas `Clinica_Atendimento` e `Clinica_Tratamento` para seus respectivos nomes completos, adicionando duas novas colunas ao dataframe:

- `Nome_Clinica_Atendimento` → Nome completo da clínica de atendimento.
- `Nome_Clinica_Tratamento` → Nome completo da clínica de tratamento.

Parâmetros

- Um dataframe contendo as colunas `Clinica_Atendimento` e `Clinica_Tratamento`.

Retorno

A função retorna o dataframe original com as colunas adicionais contendo os nomes das clínicas correspondentes.

Funcionamento

1. Verifica se as colunas `Clinica_Atendimento` e `Clinica_Tratamento` estão presentes no dataframe.
2. Utiliza um dicionário para mapear os códigos das clínicas para seus respectivos nomes completos.
3. Adiciona as colunas `Nome_Clinica_Atendimento` e `Nome_Clinica_Tratamento` ao dataframe original.
4. Exibe mensagens de progresso e finalização do processo.
5. Se as colunas `Clinica_Atendimento` ou `Clinica_Tratamento` não forem encontradas, a função interrompe a execução e retorna um erro crítico.

Mensagens Importantes

- Se as colunas forem encontradas e os nomes forem atribuídos corretamente, a função exibe:

Nomes completos das clínicas de atendimento e tratamento adicionados.

- Caso **qualquer** das colunas **Clinica_Atendimento** ou **Clinica_Tratamento** não seja encontrada, a execução é interrompida com um erro:

```
A coluna 'Clinica_Atendimento' não foi encontrada no dataframe. Função interrompida.
A coluna 'Clinica_Tratamento' não foi encontrada no dataframe. Função interrompida.
```

- Ao concluir com sucesso:

```
> Ajuste dos códigos das clínicas concluído com sucesso. Foram adicionadas as colunas
Nome_Clinica_Atendimento e Nome_Clinica_Tratamento no dataframe.
```

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Clinica_Atendimento = c(3, 25, 50, 99),
  Clinica_Tratamento = c(7, 27, 8, 99),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_clinica(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
```

#	Clinica_Atendimento	Clinica_Tratamento	Nome_Clinica_Atendimento	Nome_Clinica_Tratamento
# 1	3	7	Cabeça e Pescoço	Cirurgia Torácica
# 2	25	27	Oncologia Cirúrgica	Pediatria Oncológica
# 3	50	8	Abdômen Pélvica	Cirurgia Plástica
# 4	99	99	Sem Informação	Sem Informação

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
------	----------------	---------

A coluna 'Clinica_Atendimento' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .
A coluna 'Clinica_Tratamento' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .

A função `renomear_clinica()` facilita a interpretação dos dados do RHC ao converter códigos numéricos das clínicas para seus nomes completos. Isso melhora a legibilidade e padronização dos registros, tornando a análise mais eficiente.

Função *renomear_estadiamento_clinico()*

Descrição

Esta função mapeia os valores da coluna `Estadiamento_Clinico` para uma classificação mais geral, baseada nos principais sistemas de estadiamento de câncer. O resultado é armazenado na nova coluna:

- `Nome_Estadiamento_Clinico` → Classificação geral do estadiamento.

Parâmetros

- Um dataframe contendo a coluna `Estadiamento_Clinico`.

Retorno

A função retorna o dataframe original com a nova coluna `Nome_Estadiamento_Clinico`.

Funcionamento da Função

1. Verifica se a coluna `Estadiamento_Clinico` está presente no dataframe.
2. Mapeia os códigos de estadiamento para a nomenclatura geral com base em sistemas de classificação oncológica.
3. Adiciona a nova coluna `Nome_Estadiamento_Clinico` ao dataframe.
4. Exibe mensagens de progresso e finalização.
5. Se a coluna `Estadiamento_Clinico` não for encontrada, a função interrompe a execução e retorna um erro crítico.

Mensagens Importantes

- Se a conversão ocorrer corretamente:

Nomes do Estadiamento Clínico adicionados.

- Se a coluna **`Estadiamento_Clinico`** não for encontrada, a função **interrompe a execução** com:

A coluna 'Estadiamento_Clinico' não foi encontrada no dataframe. Função interrompida.

- Ao concluir com sucesso:

```
> Ajuste do Estadiamento Clínico concluído com sucesso. Foi adicionada uma coluna no dataframe, chamada Nome_Estadiamento_Clinico.
```

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Estadiamento_Clinico = c("0", "1A", "2B", "3C", "4A", "IV", "B"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_convertido <- renomear_estadiamento_clinico(dados_RHC)
```

Saída esperada

```
print(dados_RHC_convertido)
# Estadiamento_Clinico      Nome_Estadiamento_Clinico
# 1                0                Estádio 0
# 2               1A                Estádio I
# 3               2B                Estádio II
# 4               3C                Estádio III
# 5               4A                Estádio IV
# 6               IV                Estádio IV
# 7                B                Estádio II
```

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Estadiamento_Clinico' não foi encontrada no dataframe.	O dataframe não contém a coluna esperada.	Verifique se a coluna está presente com <code>names(dados_RHC)</code> .
Alguns valores da coluna não foram convertidos.	Pode haver códigos que não constam no mapeamento.	Adicione esses códigos ao dataframe <code>estadiamento_names</code> na função.

A função *renomear_estadiamento_clinico()* simplifica a classificação do estadiamento clínico, padronizando os valores para análise. Isso melhora a **interpretação** e a **comparação** dos estágios do câncer nos registros do RHC.

Função *ordenando_colunas()*

Descrição

Esta função reorganiza as colunas de um dataframe para que sigam uma ordem específica, definida por um conjunto de variáveis padrão. Caso alguma coluna esteja ausente, a função emite um aviso, mas continua a execução com as colunas disponíveis.

Parâmetros

- Um dataframe contendo as colunas a serem organizadas.

Retorno

A função retorna o dataframe **com as colunas reordenadas**, mantendo apenas as colunas que estiverem presentes no conjunto predefinido.

Funcionamento da Função

1. Define a ordem esperada das colunas.
2. Verifica quais colunas estão ausentes e emite um aviso com a lista de colunas faltantes.
3. Reordena as colunas conforme a lista predefinida, mantendo apenas as colunas que estão presentes no dataframe original.
4. Retorna o dataframe reorganizado.

Mensagens Importantes

- Caso **todas as colunas esperadas** estejam no dataframe:

Todas as colunas necessárias estão presentes no dataframe.

- Se **algumas colunas estiverem ausentes**, a função **continua** e exibe um alerta:

As seguintes colunas estão ausentes no dataframe:

[lista de colunas ausentes]

- Ao concluir a organização das colunas:

> Ajuste da ordem das colunas concluído com sucesso.

Exemplo de Uso

Dataframe original

```
dados_RHC <- data.frame(
  Sexo = c("M", "F", "M"),
  Idade = c(65, 50, 72),
  Tipo_Histologico = c("8000/3", "8140/3", "8070/3"),
  Estadiamento_Clinico = c("II", "I", "IV"),
  stringsAsFactors = FALSE
)
```

Executando a função

```
dados_RHC_organizado <- ordenando_colunas(dados_RHC)
```

Saída esperada

```
print(dados_RHC_organizado)
# Sexo  Idade  Tipo_Histologico  Estadiamento_Clinico
# 1  M      65           8000/3           II
# 2  F      50           8140/3           I
# 3  M      72           8070/3           IV
```

Possíveis Problemas e Soluções

Erro/Aviso	Causa Possível	Solução
Algumas colunas não aparecem na saída.	Elas estavam ausentes no dataframe original.	Verifique com <code>names(dados_RHC)</code> quais colunas estão disponíveis.
O dataframe retornado contém menos colunas do que o original.	A função mantém somente as colunas listadas no mapeamento.	Se precisar de colunas extras, edite a lista colunas dentro da função.

A função `ordenando_colunas()` garante a padronização da estrutura do dataframe, essencial para análises consistentes e integradas no contexto do Registro Hospitalar de Câncer (RHC).

Função *analise_qualidade_dos_dados()*

Descrição

A função *analise_qualidade_dos_dados()* executa uma análise completa da qualidade dos dados, verificando a integridade e a consistência das informações presentes no dataframe. Para isso, ela chama funções auxiliares que avaliam diferentes aspectos dos dados, como preenchimento de campos, códigos inexistentes, idades e anos inconsistentes, além de discrepâncias entre o tipo de câncer e o sexo do paciente.

Parâmetros

- *data*: Um dataframe contendo os dados a serem analisados.

Retorno

- A função não retorna um novo dataframe, mas exibe mensagens informativas sobre a análise realizada e eventuais inconsistências encontradas.
- Caso ocorram erros durante a execução, a função imprime mensagens de erro detalhadas.

Funcionamento

Ela executa uma série de verificações nos dados, incluindo:

1. **Análise de Completude:** Identifica valores ausentes em cada coluna.
2. **Verificação de Códigos Inconsistentes:** Confere se os códigos utilizados nos dados seguem os padrões esperados.
3. **Detecção de Problemas Diversos:** Como idades inconsistentes, anos inválidos, datas impossíveis, entre outros.

A função *analise_qualidade_dos_dados()* facilita a verificação da qualidade dos dados do RHC, ajudando a identificar problemas que podem comprometer análises estatísticas. Seu uso é essencial antes de qualquer modelagem ou interpretação dos dados.

Função *analise_completude()*

Descrição

A função *analise_completude()* calcula e avalia a completude de cada variável em um dataframe, gerando também um gráfico de barras horizontal para representar os dados ausentes. Além disso, a função classifica a qualidade da completude com base no escore de Romero & Cunha (2006).

Parâmetros

- Um dataframe contendo as variáveis a serem analisadas.

Retorno

- Retorna uma tabela com a completude de cada variável, exibindo:
 - Percentual de completude por variável
 - Número total de dados ausentes por variável
 - Classificação da completude segundo Romero & Cunha
- Gera um gráfico de barras horizontal ilustrando os dados ausentes.

Pré-requisitos

- O dataframe deve conter uma coluna chamada *Ano_do_Banco* para que seja possível calcular o intervalo temporal dos dados.
- A função verifica se a coluna *Data_Obito* está presente para tratamento específico de dados de óbito.

Funcionamento

A função executa os seguintes passos:

1. Conta o número de linhas do dataframe e verifica os anos distintos na variável *Ano_do_Banco*.
2. Calcula a quantidade de valores ausentes (NA ou "Sem informação") em cada variável.
3. Se a coluna *Data_Obito* estiver presente, calcula os casos em que há óbito registrado, mas sem data informada.
4. Converte os valores de completude para porcentagem e os classifica de acordo com Romero & Cunha:

- **Excelente:** $\geq 95\%$
 - **Bom:** 90% - 94%
 - **Regular:** 80% - 89%
 - **Ruim:** 50% - 79%
 - **Muito Ruim:** $< 50\%$
5. Exibe uma tabela com os resultados ordenados por quantidade de dados ausentes.
 6. Gera um gráfico de barras horizontal com as variáveis e seus respectivos valores ausentes.
 7. Adiciona legendas e fontes bibliográficas ao gráfico.

Mensagens Importantes

- A função exibe alertas coloridos no console para indicar o progresso da análise.
- Caso alguma coluna essencial para o cálculo esteja ausente, um erro será gerado interrompendo a execução.

Exemplo de Uso

Suponha que você tenha um dataframe chamado `dados_RHC_combinados`. Para executar a função, basta rodar:

```
# Executando a análise de completude
analise_completude(dados_RHC_combinados)
```

Se a execução for bem-sucedida:

- Será exibida uma tabela detalhando a completude das variáveis.
- O gráfico de barras horizontal será gerado automaticamente.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Ano_do_Banco' não foi encontrada.	O dataframe não contém a coluna Ano_do_Banco.	Verifique se a coluna foi gerada corretamente antes de rodar a função.
Erro ao calcular completude de Data_Obito.	A coluna Data_Obito tem valores inesperados.	Certifique-se de que os valores na coluna são coerentes (exemplo: formato de data válido).

O dataframe está vazio ou não contém variáveis válidas.	O dataframe pode estar vazio ou corrompido.	Verifique se os dados foram carregados corretamente.
---	---	--

A função *analise_completude()* permite uma avaliação eficiente da qualidade dos dados do RHC, auxiliando na detecção de inconsistências e melhorias na completude das variáveis. O uso da classificação de Romero & Cunha contribui para uma padronização da análise, facilitando a interpretação dos resultados.

Função *analise_completude_ano()*

Descrição

A função *analise_completude_ano()* calcula e avalia a completude de cada variável em um dataframe, distribuindo a proporção de completude por ano com base na variável *Ano_do_Banco*. Além disso, a função classifica a completude segundo o escore de Romero & Cunha.

Parâmetros

- Um dataframe contendo as variáveis a serem analisadas, incluindo a coluna *Ano_do_Banco*.

Retorno

- Retorna uma tabela (data.frame) que exibe a proporção de completude de cada variável por ano.
- Cada célula contém a porcentagem de completude seguida da classificação correspondente.

Pré-requisitos

- O dataframe deve conter a coluna *Ano_do_Banco*, que indica o ano dos registros.
- Os dados devem seguir um padrão em que valores ausentes são representados como NA ou "Sem informação".

Funcionamento

A função segue as seguintes etapas:

1. Verificação da Existência da Coluna *Ano_do_Banco*
 - Caso a coluna não esteja presente, a função interrompe a execução com uma mensagem de erro.
2. Identificação dos Anos Únicos
 - A função extrai todos os anos presentes na coluna *Ano_do_Banco*.
3. Cálculo da Completude por Ano
 - Para cada ano, a função verifica a quantidade de valores ausentes (NA ou "Sem informação") e calcula a taxa de completude.

4. Classificação da Completude

- A completude de cada variável é classificada conforme o escore de Romero & Cunha:
 - **E (Excelente):** $\geq 95\%$
 - **B (Bom):** $90\% - 94.9\%$
 - **R (Regular):** $80\% - 89.9\%$
 - **RU (Ruim):** $50\% - 79.9\%$
 - **MR (Muito Ruim):** $< 50\%$

5. Criação da Tabela de Resultados

- O resultado é organizado em uma tabela contendo as variáveis e suas respectivas taxas de completude por ano.

6. Exibição da Legenda

- A função exibe a legenda explicando a classificação utilizada.

7. Retorno do Dataframe

- A tabela final é retornada como um dataframe, pronto para análise e visualização.

Mensagens Importantes

- Se a coluna Ano_do_Banco não estiver presente → A função exibe um erro vermelho e interrompe a execução.
- Se não houver registros para um determinado ano → A função exibe um erro vermelho e interrompe a execução.
- Ao final, a função exibe uma mensagem verde indicando que a análise foi concluída com sucesso.

Exemplo de Uso

Suponha que você tenha um dataframe chamado `dados_RHC_combinados` com registros de diferentes anos. Você pode executar a função da seguinte forma:

```
# Executando a função e armazenando o resultado
tabela_completude_ano <- analise_completude_ano(dados_RHC_combinados)
```

Se a análise for bem-sucedida:

- `tabela_completude_ano` conterá a completude de cada variável por ano, com a classificação correspondente.

Se houver erro:

- A função interromperá a execução e informará a causa do problema.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
A coluna 'Ano_do_Banco' não foi encontrada no dataframe.	O dataframe não contém a variável Ano_do_Banco.	Verifique se a coluna existe e está corretamente nomeada.
Nenhum dado encontrado para o ano XXXX.	O dataframe não contém registros para um determinado ano.	Verifique se há dados para o ano especificado.
Erro ao calcular completude.	Alguma variável contém um tipo de dado inesperado.	Verifique se os valores ausentes estão corretamente representados como NA ou "Sem informação".

A função *analise_completude_ano()* permite uma avaliação detalhada da completude dos dados ao longo dos anos, facilitando a identificação de padrões e possíveis inconsistências. Ao utilizar um critério consolidado para classificar a qualidade dos dados, a função torna-se uma ferramenta útil para auditoria e melhoria contínua dos registros do Registro Hospitalar de Câncer (RHC).

Função *tabela_inconsistencias()*

Descrição

A função `tabela_inconsistencias()` executa múltiplas verificações de inconsistências em um dataframe e gera um relatório detalhado em formato HTML. O relatório inclui seções para diferentes tipos de inconsistências e apresenta tabelas organizadas com a contagem e proporção de registros inconsistentes.

Parâmetros

- `data`: Um dataframe contendo as colunas necessárias para todas as verificações.
- `output_file` (*opcional*): Caminho completo para o arquivo HTML onde os resultados serão salvos. O padrão é "resultados_verificacoes.html".

Retorno

- Gera um arquivo HTML contendo os resultados das verificações.
- O arquivo é automaticamente aberto no navegador padrão do sistema.

Pré-requisitos

- O dataframe deve conter as colunas relevantes para a análise de inconsistências, como códigos, datas, idades, anos e informações sobre câncer e sexo.
- As funções auxiliares para verificar inconsistências (`verificar_codigos_inconsistentes()`, `verificar_idades_inconsistentes()`, `verificar_datas_inconsistentes()`, `verificar_anos_inconsistentes()` e `verificar_cancer_sexo_inconsistentes()`) devem estar implementadas e disponíveis.

Funcionamento

A função executa as seguintes etapas:

1. Chama funções de verificação de inconsistências
 - Verifica códigos inconsistentes (valores inválidos em campos categóricos).
 - Verifica idades inválidas (valores abaixo de 0 ou acima de 150 anos).
 - Verifica datas inconsistentes (ordem cronológica incorreta entre eventos como diagnóstico, tratamento e óbito).

- Verifica anos inconsistentes (sequência temporal incoerente entre os anos de diagnóstico e tratamento).
- Verifica inconsistências entre tipo de câncer e sexo (exemplo: câncer de próstata em pacientes do sexo feminino).

2. Padroniza os nomes das colunas dos resultados

- Todas as tabelas retornadas pelas funções de verificação são formatadas com as colunas:
 - Descrição: Tipo de inconsistência detectada.
 - Quantidade: Número total de registros afetados.
 - Proporção: Percentual de registros com inconsistência.

3. Cria um relatório em HTML

- O HTML gerado contém seções separadas para cada tipo de inconsistência.
- Para cada seção, há um título, uma breve descrição do problema e uma tabela com os resultados.

4. Salva e exibe o arquivo HTML

- O relatório é salvo no caminho especificado (`output_file`).
- O arquivo é automaticamente aberto no navegador padrão.

Mensagens Importantes

- Se alguma função de verificação retornar erro → A execução continua, mas uma mensagem de aviso será exibida.
- Se não houver inconsistências detectadas → O relatório será gerado com as seções vazias.
- Se alguma das colunas obrigatórias estiver ausente → A função pode não funcionar corretamente.

Exemplo de Uso

Suponha que você tenha um dataframe chamado `dados_RHC_combinados`. Para gerar o relatório de inconsistências, execute:

```
# Gerar e abrir o relatório de inconsistências
tabela_inconsistencias(dados_RHC_combinados)
```

Se desejar salvar o relatório em um local específico:

```
# Salvar o relatório em outro diretório
tabela_inconsistencias(dados_RHC_combinados, output_file = "C:/meus_relatorios/verificacoes.html")
```

Após a execução bem-sucedida:

- O arquivo HTML será salvo no diretório especificado.
- O navegador abrirá automaticamente o relatório para visualização.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
O relatório não foi gerado.	O dataframe não contém as colunas esperadas.	Verifique se os nomes das colunas estão corretos e presentes.
Erro em uma das funções auxiliares.	Problema na execução das funções de verificação.	Execute cada função separadamente para identificar o erro.
O navegador não abriu o relatório automaticamente.	Problema na configuração do <code>browseURL()</code> .	Abra manualmente o arquivo salvo.

A função `tabela_inconsistencias()` automatiza a detecção de inconsistências nos dados do Registro Hospitalar de Câncer (RHC), facilitando a análise e garantindo a integridade das informações. O relatório HTML gerado fornece uma visão estruturada das inconsistências, auxiliando na correção e melhoria da qualidade dos dados.

Função *exibir_informacoes_finais()*

Descrição

A função *exibir_informacoes_finais()* é utilizada para monitorar automaticamente o tempo de execução do processo de construção do banco de dados. Ela exibe um resumo final contendo o tempo total decorrido, a memória utilizada e a estrutura final do dataframe processado. Essa função não está disponível para ser aplicada manualmente pelo usuário, sendo acionada exclusivamente dentro da função *construir_banco()*. Seu objetivo é fornecer um diagnóstico de desempenho durante a execução da construção do banco de dados, auxiliando no monitoramento da eficiência do processamento.

Retorno

A função exibe no console um relatório final com as seguintes informações:

- Tempo total de execução em minutos.
- Memória utilizada pelo dataframe em MB e GB.
- Total de registros e variáveis no dataframe final.

Funcionamento

A função opera da seguinte forma:

1. Cálculo do tempo total de execução

- Para medir o tempo de execução, a função utiliza *Sys.time()*, que retorna a data e a hora atuais do sistema com precisão de segundos.
- O tempo de início (*start_time_total*) é armazenado no início do processo de construção do banco e, ao final (*end_time_total*), a subtração entre esses dois valores retorna um intervalo de tempo.
- Esse intervalo pode ser expresso em segundos, minutos ou horas, permitindo calcular com precisão o tempo decorrido entre os eventos.

2. Cálculo da memória utilizada

- A função *object.size()* do R básico é utilizada para medir o espaço ocupado pelo dataframe na memória RAM, retornando o valor em bytes.
- Esse valor é convertido para MB (megabytes) ao dividi-lo por 1024^2 e para GB (gigabytes) ao dividi-lo por 1024^3 , facilitando a interpretação do consumo de memória sem a necessidade de pacotes adicionais.

3. Contagem de registros e variáveis

- A função *nrow()* conta o número de registros (linhas) no dataframe final.
- A função *ncol()* conta o número total de variáveis (colunas).
- Essas informações permitem ao usuário visualizar o tamanho do banco de dados após o processamento.

4. Exibição das informações no console

- O relatório final é formatado para destacar os valores principais em **verde**, melhorando a visualização.
- Os dados são exibidos em tempo real, garantindo um acompanhamento eficiente do processo.

5. Execução automática dentro da função *construir_banco()*

- A função *exibir_informacoes_finais()* é chamada exclusivamente ao final da execução de *construir_banco()*.
- Isso assegura que os usuários tenham um monitoramento contínuo do tempo de processamento e do uso de memória, sem precisar executar comandos adicionais.

A função *exibir_informacoes_finais()* desempenha um papel fundamental no monitoramento da construção do banco de dados, permitindo avaliar o tempo de processamento e o consumo de memória de forma automática.

Por estar integrada à função *construir_banco()*, ela assegura que o usuário tenha acesso a métricas de desempenho sem precisar de intervenção manual. Além disso, a utilização de funções do R base como *Sys.time()* e *object.size()* garante um monitoramento preciso e eficiente, sem a necessidade de pacotes adicionais. Isso contribui para um processamento mais otimizado e transparente, essencial para análises robustas de grandes bases de dados.

Função *construir_banco()*

Descrição

A função *construir_banco()* executa automaticamente todas as etapas necessárias para estruturar e preparar os dados do Registro Hospitalar de Câncer (RHC). O processo inclui a leitura dos arquivos DBF do Integrador RHC, padronização das variáveis, conversão de códigos para descrições, análise de qualidade dos dados e organização final do banco para posterior análise.

Além disso, a função monitora o tempo de execução e o uso de memória RAM, garantindo transparência e eficiência na construção do banco de dados.

Parâmetros

A função não requer parâmetros externos, pois opera automaticamente sobre os arquivos do diretório de trabalho.

Retorno

- Retorna um dataframe contendo os dados do RHC estruturados e prontos para análise.
- Se nenhum arquivo for encontrado ou ocorrer um erro, a função retorna NULL e exibe mensagens explicativas.

Pré-requisitos

- Os arquivos DBF do RHC devem estar na pasta de trabalho e seguir o padrão do Integrador RHC.
- O diretório de trabalho deve estar corretamente configurado com *getwd()*.

Funcionamento

A função *construir_banco()* executa os seguintes passos:

1. **Início do Monitoramento**
 - Registra a hora de início da execução com *Sys.time()*.
2. **Leitura dos Arquivos**
 - Chama a função *lerarquivoDBF()* para carregar e combinar os arquivos DBF.
3. **Transformação e Ajustes**
 - Renomeia e padroniza colunas: *renomear_colunas()*, *ordenando_colunas()*.

- Ajusta tipos de variáveis: *modificar_tipo_variavel()*.
- Recodifica categorias de variáveis: *recodificar_variaveis()*.

4. Conversão de Códigos

- Estados e municípios: *renomear_siglas_estados()*, *renomear_codigo_municipio_residencia()*, *renomear_codigo_municipio_hospital()*.
- CID-10 e CID-O: *renomear_CID_3digitos()*, *renomear_CID_4digitos()*.
- Outros códigos: *renomear_CNES()*, *renomear_tipo_histologico()*, *renomear_clinica()*, *renomear_estadiamento_clinico()*.

5. Análise de Qualidade dos Dados

- Verifica completude, consistência de datas e idades, e inconsistências em variáveis por meio da função *analise_qualidade_dos_dados()*.

6. Monitoramento do Tempo e Uso de Memória

- Calcula a duração total da execução subtraindo *Sys.time()* inicial do tempo final.
- Mede o consumo de memória RAM usando *object.size()*, convertendo os valores para MB ($/1024^2$) e GB ($/1024^3$).
- Exibe um relatório com o tempo total de execução, tamanho final do banco de dados e memória utilizada.

Mensagens Importantes

Situação	Mensagem
Nenhum arquivo encontrado	> Nenhum dado foi encontrado. A função será interrompida.
Erro ao carregar DBF	> Erro na leitura dos arquivos. Verifique se estão no formato correto.
Estrutura inconsistente	> As colunas dos arquivos possuem estruturas diferentes.
Banco de dados finalizado	> Construção do banco concluída com sucesso.

Exemplo de Uso

Se os arquivos DBF estiverem no diretório correto, basta executar:

```
# Construindo o banco de dados automaticamente
dados_RHC_combinados <- construir_banco()
```

Se a execução for bem-sucedida:

- dados_RHC_combinados conterá os dados organizados, prontos para análise.
- Se houver erros, a função exibirá alertas e retornará NULL.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
Nenhum arquivo rhc*.dbf encontrado.	Arquivos não estão no diretório correto ou estão com nomes incorretos.	Verifique se os arquivos estão na pasta correta e com a nomenclatura padrão.
Erro ao carregar arquivos.	Arquivo corrompido ou formato inválido.	Tente abrir os arquivos manualmente para verificar sua integridade.
Diferença no número de colunas entre arquivos.	Inconsistências na estrutura dos DBFs.	Verifique se todos os arquivos seguem o mesmo layout.
Erro no mapeamento de códigos.	Algum código pode estar ausente ou incorreto.	Revise as tabelas de referência utilizadas para conversão.

A função *construir_banco()* simplifica a preparação dos dados do RHC, garantindo que o banco esteja padronizado e pronto para análise. Ao integrar diversas etapas de processamento, conversão e verificação de qualidade, ela minimiza erros e otimiza o fluxo de trabalho dos pesquisadores.

Função *filtrar_banco()*

Descrição

A função *filtrar_banco()* permite ao usuário selecionar subconjuntos de dados do Registro Hospitalar de Câncer (RHC) com base em critérios específicos. Os filtros incluem códigos CID-3 e CID-4, anos de triagem, diagnóstico e primeira consulta, faixas etárias, tipo de caso (Analítico ou Não Analítico), sexo, estado de residência, unidade hospitalar, primeiro tratamento hospitalar, ano de início do tratamento, origem do encaminhamento e anos do banco de dados.

A função é altamente flexível e pode ser usada para refinar conjuntos de dados grandes, permitindo análises mais detalhadas e específicas.

Parâmetros

Parâmetro	Descrição	Padrão
dados	Dataframe contendo os dados do RHC.	Obrigatório
cid3digitos	Vetor de códigos CID-3 para filtrar a coluna Localizacao_Primary_3D.	NULL
cid4digitos	Vetor de códigos CID-4 para filtrar a coluna Localizacao_Primary_4D.	NULL
ano_inicio_triagem	Ano inicial para a coluna Ano_Triagem.	NULL
ano_fim_triagem	Ano final para a coluna Ano_Triagem.	NULL
ano_inicio_diagnostico	Ano inicial para a coluna Ano_Primeiro_Diagnostico.	NULL
ano_fim_diagnostico	Ano final para a coluna Ano_Primeiro_Diagnostico.	NULL
ano_inicio_primeira_consulta	Ano inicial para a coluna Ano_Primeira_Consulta.	NULL
ano_fim_primeira_consulta	Ano final para a coluna Ano_Primeira_Consulta.	NULL
idade_inicio	Idade mínima para filtrar a coluna Idade.	NULL
idade_fim	Idade máxima para filtrar a coluna Idade.	NULL

tipo_de_caso	Filtro para a coluna Tipo_de_Caso (valores possíveis: "Analítico", "Não Analítico").	NULL
sexo	Filtro para a coluna Sexo (valores possíveis: "Masculino", "Feminino").	NULL
estado_residencia	Vetor de siglas de estados para filtrar a coluna Estado_Residencia.	NULL
uf_unidade_hospital	Vetor de siglas de estados para filtrar a coluna UF_Unidade_Hospital.	NULL
primeiro_tratamento_hospital	Filtro para a coluna Primeiro_Tratamento_Hospital.	NULL
ano_inicio_tratamento	Ano inicial para filtrar a coluna Ano_Inicio_Tratamento.	NULL
ano_fim_tratamento	Ano final para filtrar a coluna Ano_Inicio_Tratamento.	NULL
origem_do_encaminhamento	Filtro para a coluna Origem_do_Encaminhamento.	NULL
ano_do_banco_inicio	Ano inicial para filtrar a coluna Ano_do_Banco.	NULL
ano_do_banco_fim	Ano final para filtrar a coluna Ano_do_Banco.	NULL

Retorno

- Retorna um dataframe filtrado contendo apenas os registros que atendem aos critérios especificados.
- Se nenhum critério for fornecido, a função retorna o dataframe original sem modificações.
- Se os critérios forem inválidos ou não corresponderem a registros, a função retorna um dataframe vazio.

Pré-requisitos

- O dataframe deve conter todas as colunas mencionadas nos critérios de filtragem.
- Os valores informados nos parâmetros devem ser compatíveis com os formatos das colunas (numérico para anos e idades, texto para categorias como sexo e estados).

Funcionamento

A função `filtrar_banco()` executa os seguintes passos:

1. Verificação da estrutura dos dados
 - Confere se todas as colunas mencionadas nos filtros existem no dataframe.
 - Valida a estrutura dos códigos CID-3 (3 caracteres) e CID-4 (4 caracteres com um ponto).
2. Aplicação dos filtros
 - Filtragem por códigos CID-3 e CID-4 (usando as colunas `Localizacao_Primary_3D` e `Localizacao_Primary_4D`).
 - Filtragem por anos de triagem, diagnóstico e primeira consulta (mantendo apenas os registros dentro do intervalo especificado).
 - Filtragem por idade (definindo um intervalo mínimo e máximo).
 - Filtragem por variáveis categóricas, como sexo, estado de residência, unidade hospitalar, primeiro tratamento e origem do encaminhamento.
3. Retorno do dataframe filtrado
 - Após aplicar os filtros, a função retorna os dados que atendem aos critérios.

Mensagens Importantes

Situação	Mensagem
Coluna necessária ausente	> Erro: O campo 'Nome_da_Coluna' não existe no dataframe.
Código CID-3 inválido	> Erro: cid3digitos deve conter exatamente 3 caracteres.
Código CID-4 inválido	> Erro: cid4digitos deve conter 4 caracteres com um ponto.
Nenhum registro encontrado	> Nenhum registro atende aos critérios informados.
Filtragem concluída	> Filtragem dos dados concluída com sucesso.

Exemplo de Uso

Suponha que você queira filtrar os dados do RHC considerando:

- Câncer de pele (CID-3 "C44").
- Diagnósticos entre 2015 e 2019.
- Pacientes entre 20 e 80 anos.
- Sexo masculino.
- Apenas registros do RJ e MG.

A chamada da função seria:

```
DADOS_FILTRADOS <- filtrar_banco(
  dados = dados_RHC_combinados,
  cid3digitos = "C44",
  ano_inicio_diagnostico = 2015,
  ano_fim_diagnostico = 2019,
  idade_inicio = 20,
  idade_fim = 80,
  sexo = "Masculino",
  estado_residencia = c("RJ", "MG")
)
```

Se os critérios forem compatíveis com os dados:

- DADOS_FILTRADOS conterá apenas os registros que atendem aos filtros especificados.
- Caso contrário, a função retorna um dataframe vazio.

Possíveis Problemas e Soluções

Erro	Causa Possível	Solução
Erro: O campo 'Ano_Primeiro_Diagnostico' não existe no dataframe.	Nome da coluna foi modificado ou não está presente.	Verifique a estrutura do dataframe e corrija o nome da coluna.
Erro: cid3digitos deve conter exatamente 3 caracteres.	Código CID-3 informado está incorreto.	Utilize apenas 3 caracteres para CID-3.
Erro: cid4digitos deve conter 4 caracteres com um ponto.	Código CID-4 informado está incorreto.	Certifique-se de incluir um ponto entre os números.
Nenhum registro retornado.	Os critérios são muito restritivos.	Tente ampliar os filtros para incluir mais registros.

A função *filtrar_banco()* facilita a seleção de registros específicos dentro do banco de dados do RHC, permitindo análises mais direcionadas. Ao oferecer múltiplas opções de filtragem, ela permite ao usuário personalizar a extração de dados de maneira eficiente e confiável.

Referência:

ROMERO, Dalia E.; CUNHA, Cynthia Braga da. Avaliação da qualidade das variáveis sócio-econômicas e demográficas dos óbitos de crianças menores de um ano registrados no Sistema de Informações sobre Mortalidade do Brasil (1996/2001). *Cadernos de Saúde Pública*, v. 22, p. 673-681, 2006.