Cursos Integrados em Vigilância em Saúde

Curso

Análise de dados para a vigilância em saúde – Curso Básico

Módulo 5 - Análises básicas de dados para vigilância em saúde - Parte II

Instituto Todos pela Saúde
ABRASCO Associação Brasileira de Saúde Coletiva
UNA-SUS Universidade Aberta do SUS
FIOCRUZ
UFPEL
UFSC

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Reitor Irineu Manoel de Souza
Vice-Reitora Joana Célia dos Passos
Pró-Reitora de Pós-graduação Werner Kraus
Pró-Reitor de Pesquisa e Inovação Jacques Mick
Pró-Reitor de Extensão Olga Regina Zigelli Garcia

CENTRO DE CIÊNCIAS DA SAÚDE

Diretor Fabrício de Souza Neves
Vice-Diretora Ricardo de Souza Magini

DEPARTAMENTO DE SAÚDE PÚBLICA

Chefe do Departamento Rodrigo Otávio Moretti Pires
Subchefe do Departamento Sheila Rúbia Lindner
Coordenadora do Curso Alexandra Crispim Boing

INSTITUTO TODOS PELA SAÚDE (ITPS)

Diretor presidente Jorge Kalil (Professor titular da Faculdade de Medicina da Universidade de São Paulo; Diretor do Laboratório de Imunologia do Incor)

ASSOCIAÇÃO BRASILEIRA DE SAÚDE COLETIVA (ABRASCO)

Presidente Rosana Teresa Onocko Campos

EQUIPE DE PRODUÇÃO

Denis de Oliveira Rodrigues
Kamila de Oliveira Belo
Marcelo Eduardo Borges
Oswaldo Gonçalves Cruz
Alexandra Crispim Boing
Antonio Fernando Boing

Módulo 5 - Análises básicas de dados para vigilância em saúde - Parte II

Curso

Análise de dados para a vigilância em saúde – Curso Básico

Dados Internacionais de Catalogação-na-Publicação (CIP)

A532 Análises básicas de dados para vigilância em saúde – Parte II/ Denis de Oliveira Rodrigues, Kamila de Oliveira Belo, Oswaldo Gonçalves Cruz, Marcelo Eduardo Borges . – Santa Catarina ; São Paulo ; Rio de Janeiro : UFSC ; ITPS ; Abrasco; 2022. 72p. (Análise de dados para a vigilância em saúde – Curso Básico; Módulo 5).

Publicação Online
10.52582/curso-analise-dados-vigilancia-modulo4-p2

1. Vigilância em saúde 2. Análise de dados I. Título

Sumário

Análises básicas de dados para vigilância em saúde

Nesse momento do curso você já conhece as ferramentas básicas para uso do *software* `R` e `RStudio` e as principais métricas para análise exploratória de dados fundamentais também para o planejamento, monitoramento e avaliação das ações de vigilância em saúde. Neste Módulo 5 aprofundaremos estas análises aprendendo a calcular alguns dos principais indicadores de saúde, apoiando o serviço de vigilância a realizar suas ações de maneira oportuna, com rapidez e segurança.

Ao final deste módulo, você será capaz de:

1. realizar cálculos de incidência e de prevalência de doenças ou agravos;
2. calcular as principais taxas para o estudo das mortalidades em saúde;
3. construir tabelas para calcular a presença ou não de uma doença e/ou agravo de saúde.

1. Calculando indicadores de saúde com software R

A análise de situação de saúde é uma atividade essencial para acompanhamento de eventos que causam ou podem causar danos à saúde da população. Mensurar esses eventos, acompanhar e construir parâmetros para avaliação são objetos centrais para tomada de decisão da vigilância em saúde, sendo seu conhecimento pré-requisito para profissionais que atuam nesta área.

Esta é a primeira etapa de um diagnóstico de saúde populacional. Conhecer como um agravo acomete a população, possibilita a caracterização de seu risco à saúde e embasa as ações de intervenção, de controle e de manejo de risco. Neste sentido, definimos risco como a probabilidade de ocorrência de um evento em uma população em um dado intervalo de tempo.

Usualmente, na Saúde Pública, essas medidas também são chamadas de indicadores e são amplamente utilizadas em diagnósticos de saúde, além de integrarem diversos painéis de monitoramento.

São exemplos de indicadores mais utilizados na rotina da área da saúde:

- **Demográficos**: fecundidade, natalidade e expectativa de vida;
- **Socioeconômico**: escolaridade e acesso ao saneamento;
- **Epidemiológicos**: morbidade e mortalidade.

Ao definirmos um indicador é importante lembrar que para ser considerado um bom indicador ele deve ser preciso e confiável. Além disso, os indicadores devem ser disponibilizados com regularidade, permitindo comparações temporais e/ou territoriais, características chaves para a rotina da vigilância em saúde.

Os indicadores podem ser expressos de duas maneiras: em **valores absolutos** ou em **valores relativos**. Veja na tabela abaixo as diferenças entre esses dois tipos:

Figura 1: Tabela de diferenças entre indicadores com valores absolutos e relativos

Indicador	Descrição	Exemplo
Valores absolutos	Representam uma contagem do número de eventos. Expressam a magnitude de um problema ou seu comportamento em curto prazo na população. São úteis para planejamento e gestão de saúde. Não permitem comparações entre populações ou na mesma população em diferentes momentos do tempo. Na vigilância em saúde são utilizados para monitorar o andamento de uma campanha de vacinação ou visualizar a magnitude de um surto, por exemplo.	Número total de óbitos; Número de casos novos de uma determinada doença; Número total de pessoas residentes em uma região geográfica
Valores relativos	Realizam o ajuste entre o número de eventos conforme o tamanho da população. Caso existam mais pessoas residindo em um município, possivelmente haverá mais eventos, assim existirá a necessidade de expressar uma relação por meio de taxa (também chamada de coeficiente) ou razão: • Razão: É a medida de frequência de um grupo de eventos relativa à frequência de outro grupo de eventos. É um tipo de fração em que o numerador e denominador são de origens distintas. • Taxa: É a relação entre o número de eventos que ocorreram em relação à uma população sob risco, em um determinado período e em uma determinada área geográfica, nos quais foi observada a frequência do evento. Informa quanto ao "risco" de ocorrência de um evento.	Razão entre o número de casos de aids no sexo masculino e o número de casos de aids no sexo feminino; incidência de dengue por 100 mil habitantes, em um município, num determinado ano

Neste módulo utilizaremos o *software* `R` para construir *scripts* com cálculo dos indicadores de saúde epidemiológicos que registram a frequência da **morbidade** e mortalidade. Estes são os mais utilizados no dia a dia para realização da análise de situação de saúde.

Acompanhe o *script* abaixo e não se preocupe com os pacotes que você ainda não conhece, eles serão explicados de forma detalhada no momento de aplicação neste módulo, replique-os em seu `RStudio`:

```
# carregando os pacotes necessários para o cálculo de indicadores epidemiológicos

if(!require(tidyverse)) install.packages("tidyverse");library(tidyverse)
if(!require(foreign)) install.packages("foreign");library(foreign)
if(!require(lubridate)) install.packages("lubridate");library(lubridate)
if(!require(epitools)) install.packages("epitools");library(epitools)
if(!require(summarytools)) install.packages("summarytools");library(summarytools)
if(!require(gtsummary)) install.packages("gtsummary");library(gtsummary)
```

2. Indicadores de morbidade

Os indicadores de morbidade apoiam o estudo da relação entre o aparecimento de doenças em uma determinada região. A morbidade frequentemente é avaliada a partir de quatro indicadores básicos: a **incidência**, a **prevalência**, a **taxa de ataque** e a **distribuição proporcional**.

Atenção

Lembre-se que um bom indicador é aquele que:

- possui uma fonte de dados confiável,
- possui um parâmetro para análise comparativa, e
- suas limitações ou vieses são conhecidos.

2.1 Incidência

O uso da incidência é muito frequente na vigilância em saúde, pois expressa o número de casos novos de uma determinada doença durante um período que se deseja estudar em uma população sob o risco de desenvolver a doença, no mesmo período.

Seu cálculo favorece a **medição da velocidade da ocorrência de uma doença na população**. Quando obtemos esta medida, sua interpretação se dá como risco de adoecimento por um determinado agravo na população. Assim, altos valores de incidências podem ser compreendidos como um alto **risco coletivo de adoecimento**.

Para esse indicador, a expressão matemática do cálculo da incidência será:

$$Incidência = \frac{número\ de\ casos\ novos\ em\ um\ determinado\ período}{número\ de\ pessoas\ sob\ risco\ no\ mesmo\ período} \times 100\ mil$$

Ao realizar este cálculo, é importante especificar o período de tempo (ano ou mês, por exemplo) analisado. Dessa forma, tanto os casos novos quanto as pessoas que já estão sob risco devem se referir ao mesmo período de tempo estudado. Além disso, o numerador do cálculo da incidência (casos novos) deve fazer parte da população que estaria exposta ao risco (denominador).

Atenção

A expressão matemática para o cálculo da incidência usualmente envolve a multiplicação por 100 mil, ou seja, expressamos a incidência comparando-a a 100.000 (100 mil) habitantes. A escolha dessa unidade de referência é arbitrária, podendo ser modificada por valores como 1.000 (1 mil) ou 10.000 (10 mil). Entretanto, essa transformação é utilizada para comparação entre municípios pequenos e grandes regiões que foram padronizados para a mesma base decimal!

Nesta subseção destacaremos os cálculos da **incidência** utilizando o banco de dados {`NINDINET.dbf`}, que é resultado do preenchimento da Ficha Individual de Notificação (FIN) do Estado de Rosas, exportada por meio do Sinan Net - Sistema de Informação de Agravos de Notificação. Acompanhe os exemplos, pratique no seu `RStudio` e vamos juntos entender como calcular.

Lembre-se que todos os bancos de dados que serão utilizados neste módulo se encontram no menu lateral "Arquivos", no módulo. Você deve fazer o *download* do material do curso diretamente da plataforma *moodle*.

Observe o *script* abaixo. Nele iremos calcular a incidência das hepatites virais por etiologia desconhecida (**CID10 = B19**) por município, apenas durante o ano de 2012, no Estado de Rosas. Para isto, acompanhe o passo a passo:

1. Primeiro vamos importar o banco de dados {`NINDINET.dbf`}, disponível no menu lateral "Arquivos", no módulo.

2. Segundo, criaremos o numerador de casos novos: utilizaremos as funções `filter()` para selecionar somente o CID B19 referente às hepatites virais notificadas no Sinan Net e também selecionar o ano de 2012 (o uso da função `year()` do pacote `lubridate` foi detalhado no Módulo 3). Também utilizamos a função `group_by()` para realizarmos a contagem dos casos por município de residência com a função `n()` dentro da `summarise()`.

Acompanhe e pratique os comandos executados no *script* abaixo:

```
# criando objeto do tipo dataframe (tabela) {`Dados`} com o banco de dados {`NINDINET.dbf`}
dados <- read.dbf(file = 'Dados/NINDINET.dbf')

# Criando um novo objeto a partir da tabela {`Dados`}
tabela_inc_hepatite <- dados |>

  # Utilizando a função filter() para diversos critérios de filtragem de dados
  filter(

    # Filtrando os agravos de hepatites virais (código "B19")
    ID_AGRAVO == "B19",

    # Filtrando apenas casos com ano de primeiros sintomas igual a 2012
    year(DT_SIN_PRI) == 2012) |>

  # Removendo as categorias (levels) ausentes na coluna ID_AGRAVO filtrada pelo
# uso da função droplevels()
  mutate(ID_AGRAVO = droplevels(ID_AGRAVO)) |>

  # Agrupando por município de residência
  group_by(ID_MN_RESI) |>

  # Contanto o número de casos novos por município de residência
  summarise(N_CASOS_NOVOS = n())

# Visualizando a tabela resultante
tabela_inc_hepatite
```

```
#> # A tibble: 14 × 2
#>    ID_MN_RESI N_CASOS_NOVOS
#>         <int>         <int>
#>  1     610045             2
#>  2     610170             7
#>  3     610213            72
#>  4     610260             1
#>  5     610270             1
#>  6     610285             1
#>  7     610320             1
#>  8     610330             3
#>  9     610350             5
#> 10     610414             1
#> 11     610490             3
#> 12     610510             4
#> 13     610520             1
#> 14     610580             1
```

Observe o *output*: como resultado, você verá uma tabela com duas colunas. A primeira coluna (`ID_MN_RESI`) indica o código de cada município e a segunda coluna (`N_CASOS_NOVOS`) indica o número de casos em cada município.

3. No terceiro passo vamos utilizar a população dos municípios do Estado de Rosas como nosso denominador para que possamos calcular a incidência de hepatites virais de etiologia desconhecida. Para isso, importaremos a tabela de população por município, disponível no menu lateral "Arquivos", no módulo.

Acompanhe o *script* abaixo e replique em seu `RStudio`:

```
# Criando a tabela "tabela_populacao" (dataframe) a partir do arquivo
# "tabela.csv", que contém o tamanho populacional por município de Rosas
tabela_populacao <- read_csv2("Dados/tabela.csv")

# Visualizando a tabela carregada
tabela_populacao
```

```
#> # A tibble: 14 × 2
#>    cod_mun POPULACAO
#>      <dbl>     <dbl>
#>  1  610045     15009
#>  2  610170     69189
#>  3  610213    201452
#>  4  610260     30205
#>  5  610270     19688
#>  6  610285     20448
#>  7  610320      6713
#>  8  610330     17559
#>  9  610350     10018
#> 10  610414      4164
#> 11  610490     79524
#> 12  610510     32042
#> 13  610520     43143
#> 14  610580      3434
```

Observe que na primeira coluna desta tabela (`cod_mun`), você verá novamente o código de cada município. Já na segunda coluna (`POPULACAO`) temos os valores referentes ao número de habitantes em cada município de Rosas.

Lembre-se de que é uma boa prática ao usar uma linguagem de programação ir criando novos objetos e salvando suas tabelas de análises!

Cada objeto criado armazenará seus dados e permitirá que estes sejam acessados no momento que desejar e inclusive podendo ser reutilizáveis, tornando sua análise mais rápida e segura. Caso tenha dúvida em como criar objetos acesse o Módulo 2 deste curso.

4. No quarto passo, vamos agregar a população para o cálculo utilizando a função `left_join()`. Isto porque, para o cálculo da incidência, precisaremos unir a tabela de casos novos e a tabela de população. Para isso utilizaremos como chave única para cruzamento o código do município, variável comum às duas tabelas. Para isto, no argumento `by` dentro da função `left_join()` incluímos as variáveis `"ID_MN_RESI"` e `"cod_mun"`.

Veja abaixo como deve ser feito o *script* e replique no seu `RStudio`:

```
# Unindo a tabela "tabela_inc_hepatite" com "tabela_populacao", pela coluna com
# o código do município em cada uma
tabela_inc_hepatite |>

  # unindo a tabala de casos de hepatites virais com a tabela de população
  # utilizando os códigos do municípios de cada tabela
  left_join(tabela_populacao, by = c("ID_MN_RESI" = "cod_mun")) |>

  # Criando uma nova coluna com a incidência de casos por 100000 habitantes com
  # a função mutate()
  mutate(INCIDENCIA = N_CASOS_NOVOS / POPULACAO * 100000)
```

```
#> # A tibble: 14 × 4
#>    ID_MN_RESI N_CASOS_NOVOS POPULACAO INCIDENCIA
#>         <dbl>         <int>     <dbl>      <dbl>
#>  1     610045             2     15009      13.3
#>  2     610170             7     69189      10.1
#>  3     610213            72    201452      35.7
#>  4     610260             1     30205       3.31
#>  5     610270             1     19688       5.08
#>  6     610285             1     20448       4.89
#>  7     610320             1      6713      14.9
#>  8     610330             3     17559      17.1
#>  9     610350             5     10018      49.9
#> 10     610414             1      4164      24.0
#> 11     610490             3     79524       3.77
#> 12     610510             4     32042      12.5
#> 13     610520             1     43143       2.32
#> 14     610580             1      3434      29.1
```

Observe no *output* que a nova tabela criada possui quatro colunas:

- `ID_MN_RESI` indica o código do município,
- `N_CASOS_NOVOS` demonstra o total de casos novos,
- `POPULACAO` contém o tamanho populacional e
- `INCIDENCIA` apresenta o índice que queremos: a incidência de casos a cada 100 mil habitantes.

Na avaliação dos municípios do Estado de Rosas é possível verificar também que o município 610350 (Flor de Lis) possui a maior incidência, 35,7 casos por 100 mil habitantes, enquanto o município 610520 (Lótus) possui a menor incidência, com 2,3 por 100 mil habitantes no Estado.

Observe também os municípios 610330 (Glicínia) e o 610490 (Zínia) possuem o mesmo número de casos: três casos, porém suas valores de incidência bastante distintas (17,1 e 3,77 por 100 mil habitantes, respectivamente), isto ocorre pois o tamanho de suas populações é distinto!

Perceba que utilizamos sempre o operador *pipe* (`|>`) para indicar uma sequência de ações. Caso encontre erros para executar seu código, faça revisão de toda a escrita desde a ortografia, até a escrita de pontos e inclusão de símbolos. Lembre-se sempre que o comando somente será executado corretamente se o *script* for escrito da maneira correta.

Para buscar ajuda, cole o erro ou o warning no seu navegador da internet (consulte o google ou o *Stack Overflow*).

2.2 Prevalência

A prevalência é o indicador mais utilizado para **mensurar a frequência de doenças crônicas de longa duração**. Isto porque seu numerador se refere ao total de pessoas que se apresentam doentes no período analisado. Na prevalência são somados os casos novos e os casos já conhecidos. A sua interpretação não se refere a risco de adoecer, e sim **o quanto uma doença persiste numa população**.

Para calcular esse indicador, a expressão matemática do cálculo da prevalência é:

$$Prevalência = \frac{número\ de\ casos\ existentes\ em\ um\ determinado\ período}{número\ total\ de\ pessoas\ no\ mesmo\ período} \times 100\ mil$$

Atenção

A expressão matemática para o cálculo da prevalência usualmente envolve a multiplicação por 100 mil, ou seja, expressamos a prevalência comparando por 100.000 (100 mil) habitantes. A escolha dessa unidade de referência é arbitrária, podendo ser modificada por valores como 1.000 (1 mil) ou 10.000 (10 mil). Entretanto, essa transformação é utilizada para comparação entre municípios pequenos e grandes regiões que foram padronizados para a mesma base decimal!!

Agora vamos praticar o cálculo deste indicador utilizando os dados referentes à prevalência de hanseníase no **Estado do Acre** durante o ano de 2021. Este banco de dados se encontra no menu lateral "Arquivos" no módulo!

A prevalência de hanseníase estima a magnitude da endemia no Acre. Ela deve ser estudada com base na totalidade de casos em tratamento no momento em que realizamos a sua avaliação. Uma alta prevalência (valores acima de 5 casos por 10 mil habitantes) pode indicar um cenário de baixo desenvolvimento socioeconômico e falta de ações efetivas do município para o controle da doença. Por outro lado, a baixa prevalência (valores menores que 1 caso por 10 mil habitantes) pode indicar que a hanseníase não deve considerada um problema de saúde pública.

Nesta análise, o profissional de vigilância necessitará importar o arquivo {`base_hans_ac.dbf`} no formato ".dbf" dos casos de hanseníase no Acre, disponível no menu lateral "Arquivos", no módulo. Lembre-se que a importação de arquivos nesse formato é feita pela função `read.dbf()` do pacote `foreign`.

Acompanhe abaixo o script e replique o código em seu `RStudio`:

```
# criando objeto do tipo data.frame {`base_hans`} que armazenará o banco
# de dados de hanseníase do Estado do Acre

base_hans <- read.dbf(file = 'Dados/base_hans_ac.dbf')
```

Como vamos avaliar o ano de 2021, vamos selecionar um recorte de dois anos anteriores (2019 e 2020) e, assim, incluir os casos em tratamento que podem ter iniciado o esquema terapêutico nesses anos, mas se mantêm em andamento no ano de avaliação. Dessa forma, vamos filtrar os casos abertos diagnosticados entre 2019 e 2021. Acompanhe os passos:

1. Primeiro, vamos criar uma nova variável, extraindo o ano da data de diagnóstico utilizando a função `year()` do pacote `lubridate` dentro da função `mutate()`.

2. Em seguida, vamos filtrar os casos diagnosticados entre 2019 e 2021 e, simultaneamente, vamos selecionar os casos que ainda estão em tratamento, ou seja, aqueles cuja variável tipo de saída (`TPALTA_N`) ainda não foi preenchida (está nula ou em branco). Para filtrar os casos abertos, vamos utilizar a função `is.na()` para selecionar os registros com a variável `TPALTA_N` em branco, dentro da função `filter()`.

3. Por fim, vamos contar os casos confirmados de hanseníase com a função `count()`. Mas não vamos indicar nenhuma variável dentro da função, pois queremos a contagem total do filtro realizado antes.

Acompanhe o código abaixo e reproduza no seu `RStudio`:

```
base_hans |>

 # Criando uma nova coluna com o ano de diagnóstico com a função mutate() e year()
 mutate(ano_diag = year(DT_DIAG)) |>

 # Filtrando os casos diagnosticados entre 2019 e 2021 e "tipo de saída"
 # (TPALTA_N) com valores em branco
 filter(ano_diag >= 2019, ano_diag <= 2021, is.na(TPALTA_N)) |>

 # Contanto o número de casos por ano de diagnóstico com a função count()
 count()
```

```
#>     n
#> 1 171
```

Visualize que obtivemos um *output* apenas com um único valor, indicando o total de casos em branco neste conjunto de dados (`n`), ou seja, temos 171 casos que ainda estão em tratamento no Estado do Acre em 2021.

4. Mas para o cálculo da prevalência precisamos da população estimada do Estado do Acre no mesmo ano. Conseguimos essa população fazendo a tabulação no site do Datasus e você pode importá-la por meio do arquivo {`pop_ac_09_21.csv`}. Lembre-se de copiar o caminho do arquivo localizado no menu lateral "Arquivos", do módulo.

Acompanhe os comandos escritos no *script* abaixo e replique:

```
# Criando o objeto {`pop_ac`} com a população estimada para o Acre
# O argumento `col_types = list("character")` indica que todas as
# colunas serão lidas como strings de texto (character)
pop_ac <- read_csv2('Dados/pop_ac_09_21.csv', col_types = list("character"))
```

Perceba que para a construção do código executado no *script* acima incluímos o argumento chamado `col_types`. Ele foi necessário para garantir que a primeira coluna do arquivo, chamada "Codigo", mantenha-se no formato original do tipo character.

Agora vamos visualizar como ficou a tabela que criamos para armazenar a população por ano da estimativa segundo município do Acre. Digite o *script* abaixo e replique em seu `RStudio`:

```
# Visualizando as linhas e colunas do dataframe {`pop_ac`}
glimpse(pop_ac)
```

```
#> Rows: 22
#> Columns: 15
#> $ Codigo    <chr> "120001", "120005", "120010", "120013", "120017", "120020", …
#> $ Municipio <chr> "Acrelândia", "Assis Brasil", "Brasiléia", "Bujari", "Capixa…
#> $ `2009`    <dbl> 12769, 6180, 21758, 8633, 8812, 80978, 15373, 33678, 6653, 1…
#> $ `2010`    <dbl> 13081, 6335, 22325, 8838, 9179, 81907, 15754, 33816, 6862, 1…
#> $ `2011`    <dbl> 13327, 6457, 22771, 8999, 9468, 82638, 16054, 33925, 7026, 1…
#> $ `2012`    <dbl> 13579, 6583, 23231, 9165, 9764, 83389, 16363, 34037, 7195, 1…
#> $ `2013`    <dbl> 13821, 6703, 23670, 9324, 10048, 84109, 16658, 34144, 7357, …
#> $ `2014`    <dbl> 14069, 6826, 24120, 9486, 10339, 84845, 16960, 34254, 7523, …
#> $ `2015`    <dbl> 14318, 6951, 24574, 9650, 10632, 85587, 17265, 34364, 7690, …
#> $ `2016`    <dbl> 14551, 7067, 24996, 9803, 10905, 86279, 17550, 34467, 7845, …
#> $ `2017`    <dbl> 14781, 7181, 25414, 9954, 11175, 86963, 17830, 34569, 7999, …
#> $ `2018`    <dbl> 15020, 7300, 25848, 10111, 11456, 87673, 18122, 34675, 8159,…
#> $ `2019`    <dbl> 15256, 7417, 26278, 10266, 11733, 88376, 18411, 34780, 8317,…
#> $ `2020`    <dbl> 15490, 7534, 26702, 10420, 12008, 89072, 18696, 34884, 8473,…
#> $ `2021`    <dbl> 15721, 7649, 27123, 10572, 12280, 89760, 18979, 34986, 8628,…
```

Nesta tabela, você verá na primeira linha o código do município, na segunda linha o nome do município, e a partir da terceira linha as estimativas populacionais para cada um dos municípios por ano entre 2009 e 2021.

5. Como vamos precisar apenas do ano de 2021, do estado todo, vamos utilizar a função `summarise()` para resumir e somar toda a população dos municípios usando a função `sum()`. Perceba que nas colunas da tabela {`pop_ac`} os anos da população estimada estão organizados em colunas. Para somar somente o ano de 2021, vamos nos referenciar à coluna utilizando crases (assim: `` `2021` ``). Isso é necessário para que o `R` possa entender que estamos nos referindo a uma coluna de um *dataframe* e não um valor numérico (o título da coluna é 2021). Abaixo vamos criar a coluna com a população total de 2021. Replique o *script* abaixo em seu `RStudio`:

```
pop_ac |>

# Criando a coluna com a população total do Estado do Acre em
# 2021 com a função summarise() e sum()
  summarise(pop_total_ac_2021 = sum(`2021`))
```

```
#> # A tibble: 1 × 1
#>   pop_total_ac_2021
#>               <dbl>
#> 1            906876
```

Você deve observar que a população do Acre, portanto, é de 906.876 habitantes em 2021.

6. Por fim, calcularemos a prevalência de hanseníase no Acre em 2021 por 10 mil habitantes. Para isto, iremos substituir os valores na expressão matemática a seguir, veja:

$$Prevalência = \frac{171}{906876} \times 10.000$$

Observe como fazemos agora no `R`, escreva conforme *script* abaixo a operação matemática no seu `RStudio`.

```
# Divisão do número de casos pela população do Acre, seguida da multiplicação com 10000
(171 / 906876) * 10000
```

```
#> [1] 1.885594
```

O resultado desta operação é o valor de 1.885594. Desta forma, a prevalência dos casos em tratamento no Estado do Acre em 2021 é de 1,89 pessoas a cada 10 mil habitantes.

7. Agora, vamos calcular este indicador para todos os municípios do estado. Para tal, vamos repetir o comando utilizado anteriormente, mas indicando dentro da função `count()` o município de residência atual (`MUNIRESAT`). Para fins de reprodutibilidade, iremos salvar o resultado da contagem por municípios em um novo objeto, que chamaremos de {`casos_hans_ac_21`}. Veja o *script* abaixo e repita em seu `RStudio`:

```
# Criando uma tabela (dataframe) com o nome casos_hans_ac_21
casos_hans_ac_21 <- base_hans |>

  # Filtrando os casos na coluna "tipo de saída" (TPALTA_N) com valores em branco
  filter(is.na(TPALTA_N)) |>

  # Contando o número de casos com valores em branco por município
  count(MUNIRESAT)

# Visualizando a tabela resultante
casos_hans_ac_21
```

```
#>    MUNIRESAT  n
#> 1     120001  2
#> 2     120005  3
#> 3     120010  8
#> 4     120013  3
#> 5     120017  3
#> 6     120020 12
#> 7     120025  9
#> 8     120030  6
#> 9     120032  1
#> 10    120033  2
#> 11    120035  1
#> 12    120038  3
#> 13    120039  5
#> 14    120040 70
#> 15    120042  2
#> 16    120043  1
#> 17    120045  6
#> 18    120050  7
#> 19    120060 11
#> 20    120070 10
#> 21    120080  5
#> 22      <NA>  1
```

A tabela resultante apresentará na primeira coluna o código do município de residência atual, enquanto a segunda coluna irá indicar o total de casos em cada um destes municípios.

Para seguir nesta etapa precisamos da população de cada município para o cálculo da prevalência. Então, seguiremos os passos abaixo:

1. Primeiro, utilizaremos novamente o objeto {`pop_ac`} criado para armazenar a população estimada para o Acre.

2. Segundo, iremos unir a tabelas população e a tabela de casos em tratamento utilizando a função `left_join()`; para uni-las, precisamos indicar qual coluna será chave de ligação entre as duas tabelas utilizando o argumento `by`, ou seja, incluímos a variável `MUNIRESAT` na tabela de casos e a variável `Codigo` da tabela de população.

Acompanhe o *script* abaixo e repita em seu `RStudio`:

```
# Criando uma tabela (dataframe) com o nome {`prev_casos_ac`}
prev_casos_ac <- casos_hans_ac_21 |>

  # Juntando a tabela casos_hans_ac_21 com pop_ac, pelas colunas com os códigos de município de cada uma
  left_join(pop_ac, by = c("MUNIRESAT" = "Codigo"))

# Visualizando a tabela resultante
prev_casos_ac
```

```
#>    MUNIRESAT  n            Municipio   2009   2010   2011   2012   2013   2014
#> 1     120001  2           Acrelândia  12769  13081  13327  13579  13821  14069
#> 2     120005  3         Assis Brasil   6180   6335   6457   6583   6703   6826
#> 3     120010  8           Brasiléia  21758  22325  22771  23231  23670  24120
#> 4     120013  3              Bujari   8633   8838   8999   9165   9324   9486
#> 5     120017  3             Capixaba   8812   9179   9468   9764  10048  10339
#> 6     120020 12      Cruzeiro do Sul  80978  81907  82638  83389  84109  84845
#> 7     120025  9       Epitaciolândia  15373  15754  16054  16363  16658  16960
#> 8     120030  6                Feijó  33678  33816  33925  34037  34144  34254
#> 9     120032  1              Jordão   6653   6862   7026   7195   7357   7523
#> 10    120033  2          Mâncio Lima  15417  15864  16216  16577  16924  17277
#> 11    120035  1 Marechal Thaumaturgo  14265  14843  15298  15765  16213  16670
#> 12    120038  3    Plácido de Castro  17695  17954  18158  18368  18569  18774
#> 13    120039  5         Porto Walter   9227   9573   9845  10125  10393  10667
#> 14    120040 70           Rio Branco 342445 350589 356998 363589 369899 376348
#> 15    120042  2      Rodrigues Alves  14450  15012  15455  15910  16345  16791
#> 16    120043  1  Santa Rosa do Purus   4658   4894   5080   5271   5454   5642
#> 17    120045  6     Senador Guiomard  20770  21053  21276  21505  21724  21948
#> 18    120050  7       Sena Madureira  38790  39676  40373  41090  41777  42478
#> 19    120060 11             Tarauacá  36351  37131  37745  38377  38981  39599
#> 20    120070 10               Xapuri  16424  16788  17074  17369  17651  17939
#> 21    120080  5           Porto Acre  15096  15524  15861  16207  16538  16877
#> 22      <NA>  1                <NA>     NA     NA     NA     NA     NA     NA
#>       2015   2016   2017   2018   2019   2020   2021
#> 1    14318  14551  14781  15020  15256  15490  15721
#> 2     6951   7067   7181   7300   7417   7534   7649
#> 3    24574  24996  25414  25848  26278  26702  27123
#> 4     9650   9803   9954  10111  10266  10420  10572
#> 5    10632  10905  11175  11456  11733  12008  12280
#> 6    85587  86279  86963  87673  88376  89072  89760
#> 7    17265  17550  17830  18122  18411  18696  18979
#> 8    34364  34467  34569  34675  34780  34884  34986
#> 9     7690   7845   7999   8159   8317   8473   8628
#> 10   17635  17968  18297  18638  18977  19311  19643
#> 11   17132  17563  17988  18430  18867  19299  19727
#> 12   18982  19175  19366  19565  19761  19955  20147
#> 13   10944  11201  11456  11720  11982  12241  12497
#> 14  382864 388932 394924 401155 407319 413418 419452
#> 15   17241  17660  18074  18504  18930  19351  19767
#> 16    5831   6007   6181   6362   6540   6717   6893
#> 17   22174  22385  22593  22810  23024  23236  23446
#> 18   43187  43847  44499  45177  45848  46511  47168
#> 19   40224  40805  41379  41976  42567  43151  43730
#> 20   18230  18501  18769  19048  19323  19596  19866
#> 21   17219  17538  17853  18180  18504  18824  19141
#> 22      NA     NA     NA     NA     NA     NA     NA
```

Perceba que a tabela resultante irá conter, além do código do município de residência atual e o número de casos por município, uma nova coluna indicando o nome do município, e diversas colunas indicando a estimativa do tamanho populacional destes municípios para diferentes anos.

3. O terceiro passo será calcular a prevalência da hanseníase por município do Estado do Acre, criando uma nova coluna com o resultado do cálculo utilizando a função `mutate()`.

4. Iremos selecionar somente as variáveis que nos importam nesta tabela (`MUNIRESAT`, `Municipio`, `prevalencia_2021`). Acompanhe o script abaixo e replique-o em seu `RStudio`:

```
prev_casos_ac |>

  # Criando uma coluna com o cálculo de prevalência da população por 10 mil
  # habitantes em 2021 com a função mutate()
  mutate(prevalencia_2021 = (n / `2021`) * 10000) |>

  # Selecionando apenas as colunas MUNIRESAT, Municipio e prevalencia_2021 com
  # a função select()
  select(MUNIRESAT, Municipio, prevalencia_2021)
```

```
#>    MUNIRESAT            Municipio prevalencia_2021
#> 1     120001           Acrelândia        1.2721837
#> 2     120005         Assis Brasil        3.9220813
#> 3     120010            Brasiléia        2.9495262
#> 4     120013               Bujari        2.8376844
#> 5     120017             Capixaba        2.4429967
#> 6     120020      Cruzeiro do Sul        1.3368984
#> 7     120025       Epitaciolândia        4.7420834
#> 8     120030                Feijó        1.7149717
#> 9     120032              Jordão        1.1590172
#> 10    120033          Mâncio Lima        1.0181744
#> 11    120035 Marechal Thaumaturgo        0.5069195
#> 12    120038    Plácido de Castro        1.4890554
#> 13    120039         Porto Walter        4.0009602
#> 14    120040           Rio Branco        1.6688441
#> 15    120042      Rodrigues Alves        1.0117873
#> 16    120043  Santa Rosa do Purus        1.4507471
#> 17    120045     Senador Guiomard        2.5590719
#> 18    120050       Sena Madureira        1.4840570
#> 19    120060             Tarauacá        2.5154356
#> 20    120070               Xapuri        5.0337260
#> 21    120080           Porto Acre        2.6121937
#> 22      <NA>                <NA>               NA
```

Observe que a tabela resultante irá apresentar, além das informações com o código e nome do município, os valores de prevalência para cada um deles no ano de 2021. Observe no *output* acima que a prevalência da hanseníase no Estado do Acre varia de 0,5 por 10 mil habitantes no município de Marechal Thaumaturgo até 5 por 10 mil habitantes no município de Xapuri.

Agora, vamos avançar um pouco mais e aprender a calcular com o *software* `R` alguns indicadores de mortalidade!

3. Indicadores de Mortalidade

Os indicadores de mortalidade são amplamente utilizados na vigilância em saúde e em diversas outras áreas para além da saúde. O Sistema de Informação de Mortalidade (SIM) é um dos sistemas de informações no SUS focado na melhoria das ações em saúde. É um sistema com boa cobertura, preenchimento e completitude. Algumas de suas principais vantagens são:

- ter seus dados regularmente coletados;
- possuir pontos de coleta, ou seja, capilaridade em todo país;
- ter um padrão de preenchimento nacional, e
- possuir uma longa série histórica de coleta de dados.

Desta forma, existem vários indicadores de mortalidade que integram as análises de saúde. Nesta subseção vamos revisar alguns deles: **taxa bruta de mortalidade, taxas específicas de mortalidade por idade e sexo, mortalidade proporcional por grupo de causas, taxa de mortalidade infantil e taxa de mortalidade materna**. Todos serão calculados utilizando os dados de óbitos entre 2015 e 2020 do Estado do Acre.

Vamos lá! Comece importando o arquivo {`do_ac.dbf`} para o `R` utilizando a função `read.dbf()`. Também importaremos vários arquivos necessários para os cálculos dos indicadores, são eles:

- Banco de dados {`CID-10-GRUPOS.CSV`} que contém variáveis sobre classificação internacional de doenças por grupos de causas (capítulos);
- Banco de dados {`pop_ac_09_21.csv`} que contém a estimativas da população total do Estado do Acre de 2009 a 2021;
- Banco de dados {`pop_ac_sexo_idade_20.csv`} que contém a população do Estado do Acre em 2020 estratificada por sexo e faixa etária e;
- Banco de dados {`nv_ac_15_20.csv`}, com dados de nascidos vivos no Estado do Acre entre 2015 e 2020.

Todos os dados foram importados do repositório do Datasus e estão disponíveis localizado no menu lateral “Arquivos”, do módulo.

```
# Importando diferentes banco de dados para o R
base_obito_ac <- read.dbf(file = 'Dados/do_ac.dbf')
grupos_causas <- read_csv2(file = 'Dados/CID-10-GRUPOS.csv')
pop_ac <- read_csv2(file = 'Dados/pop_ac_09_21.csv', col_types = list("character"))
pop_ac_sexo_idade <- read_csv2(file = 'Dados/pop_ac_sexo_idade_20.csv')
nv_ac <- read_csv2(file = 'Dados/nv_ac_15_20.csv')
```

Atenção

Fique atento ao caminho escolhido para armazenar os bancos de dados em seu computador. Caso tenha colocado-as em outro diretório ou pasta, escreva o caminho deste arquivo para que o `R` possa importá-lo para sua análise.

Para os cálculos de indicadores de mortalidade, devido a várias características dos métodos de cálculos, algumas transformações do banco de dados são oportunas e devem serem feitas de imediato. Isso é para facilitar a manipulação sempre que forem realizar alguma análise com bancos de dados. Uma das tarefas mais importante é criar um conjunto de novas variáveis.

Acompanhe a tabela abaixo que apresenta as variáveis que serão criadas e a descrição de cada uma delas, com as alterações que realizaremos para facilitar nossas análises:

Figura 2: Tabela com as alterações realizadas no banco de dados de óbitos.

Variável	Descrição da alteração efetuada
CAUSA_BASICA_3C	Nova variável com a causa básica do óbito com três caracteres
DTOBITO	Variável DTOBITO transformada para formato Date
ANO_OBITO	Nova variável com apenas o ano do óbito
DTNASC	Variável DTNASC transformada para formato Date
SEXO	Variável SEXO decodificada
RACACOR	Variável RACACOR decodificada
UNIDADE_IDADE	Nova variável indicada qual unidade de idade (se anos, meses, dias)
IDADE_NUM	Nova variável com idade no formato Numeric
FX_ETARIA_11C	Nova variável com categorização da idade em 11 classes
FAIXA_ETARIA_M1ANO	Nova variável com categorização de menores 1 ano
LOCOCOR	Variável LOCOCOR decodificada

Observe que realizaremos de uma só vez todas essas alterações apontadas, escrevendo diversas linhas de código. Caso sinta-se mais seguro, você poderá realizar as mesmas alterações variável por variável, de forma mais confortável, construindo seu estilo de trabalho. Entretanto, isso implicará em um maior tempo de escrita do seu *script*.

Agora observe no passo a passo abaixo como colocaremos em prática estas transformações:

1. São utilizados no código as funções `mutate()`, `str_sub()`, `dmy()`, `year()`, `case_when()` e `factor()`. Lembre-se que na função `case_when()`, quando nenhum dos critérios é atendido, usamos `TRUE ~ NA_character_` para indicar que estes valores devem ser convertidos em `NA`. Caso tenha mais alguma dúvida, consulte os módulos anteriores do curso.

Veja o *script* abaixo com muita atenção, leia os comentários explicando cada parte e replique-o no seu `RStudio`:

```
# Criando uma tabela (dataframe) com o nome base_obito_ac
base_obito_ac <- base_obito_ac |>

# Criando novas colunas com a função mutate()
  mutate(

  # Criando coluna com extração de apenas os 3 primeiros caracteres
  # da coluna CAUSABAS pelo uso da função str_sub()
    CAUSA_BASICA_3C = str_sub(CAUSABAS, 1, 3),

  # Convertendo as informações na coluna DTOBITO para o formato de
  # data (DD-MM-YYYY) com a função dmy()
    DTOBITO = dmy(DTOBITO),

  # Obtendo apenas o ano de óbito a partir da coluna DTOBITO com a função year()
    ANO_OBITO = year(DTOBITO),

  # Convertendo as informações na coluna DTNASC para o formato de
  # data (DD-MM-YYYY) com a função dmy()
    DTNASC = dmy(DTNASC),

  # Transformando os códigos de sexo nos nomes correspondentes com a
    # função case_when()
    SEXO = case_when(
      SEXO == "1" ~ "masculino",
      SEXO == "2" ~ "feminino",
      TRUE ~ NA_character_
    ),

  # Convertendo a coluna SEXO em uma variável do tipo factor
    SEXO = factor(SEXO, levels = c("masculino", "feminino")),


   # Transformando os códigos de raça/cor nos nomes correspondentes com a
  # função case_when()
    RACACOR = case_when(
      RACACOR == "1" ~ "Branca",
      RACACOR == "2" ~ "Preta",
      RACACOR == "3" ~ "Amarela",
      RACACOR == "4" ~ "Parda",
      RACACOR == "5" ~ "Indígena",
      TRUE ~ NA_character_
    ),
```

```
  # Transformando os códigos de unidade da idade nos nomes correspondentes com a
  # função case_when(). A classificação utiliza apenas o primeiro caractere da
  # coluna IDADE, com uso da função str_sub()
     UNIDADE_IDADE = case_when(
       (str_sub(IDADE, 1, 1) == "5") ~ "anos+",
       (str_sub(IDADE, 1, 1) == "4") ~ "anos",
       (str_sub(IDADE, 1, 1) == "3") ~ "meses",
       (str_sub(IDADE, 1, 1) == "2") ~ "dias",
       (str_sub(IDADE, 1, 1) == "1") ~ "horas",
       (str_sub(IDADE, 1, 1) == "0") ~ "minutos"
     ),

  # Extraindo os dois últimos dígitos da idade com a função str_sub() e
  #  convertendo para número utilizando a função as.numeric()
     IDADE_NUM = as.numeric(str_sub(IDADE, 2, 3)),

  # Criando uma classificação de faixa etária por intervalos utilizando
  # as variáveis criadas acima
     FX_ETARIA_11C = case_when(
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 0, 4)) ~ "0 a 4",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 5, 9)) ~ "5 a 9",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 10, 14)) ~ "10 a 14",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 15, 19)) ~ "15 a 19",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 20, 29)) ~ "20 a 29",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 30, 39)) ~ "30 a 39",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 40, 49)) ~ "40 a 49",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 50, 59)) ~ "50 a 59",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 60, 69)) ~ "60 a 69",
       (UNIDADE_IDADE == "anos" & between(IDADE_NUM, 70, 79)) ~ "70 a 79",
       ((UNIDADE_IDADE == "anos" & IDADE_NUM >= 80) | UNIDADE_IDADE ==
"anos+") ~ "80 e mais",
       TRUE ~ NA_character_
     ),
  # Convertendo a coluna FX_ETARIA_11C em uma variável do tipo factor
     FX_ETARIA_11C = factor(
       FX_ETARIA_11C,
       levels = c(
         "0 a 4"  , "5 a 9"  , "10 a 14", "15 a 19", "20 a 29", "30 a 39",
         "40 a 49", "50 a 59", "60 a 69", "70 a 79", "80 e mais"
       )
     ),
```

```
  # Reclassificando variáveis da coluna FX_ETARIA_M1ANO para os casos em que
  # a idade seja menor do que 1 ano
     FX_ETARIA_M1ANO = case_when(
       (UNIDADE_IDADE == "anos" & IDADE_NUM < 1) ~ "< 1 ano",
       (UNIDADE_IDADE == "dias") ~ "< 1 ano",
       (UNIDADE_IDADE == "meses") ~ "< 1 ano",
       (UNIDADE_IDADE == "horas") ~ "< 1 ano",
       (UNIDADE_IDADE == "minutos") ~ "< 1 ano",
       TRUE ~ NA_character_
     ),

  # Transformando os códigos de local de ocorrência nos nomes correspondentes
  # com a função case_when()
     LOCOCOR = case_when(
       LOCOCOR == "1" ~ "Hospital",
       LOCOCOR == "2" ~ "Outros estab saude",
       LOCOCOR == "3" ~ "Domicilio",
       LOCOCOR == "4" ~ "Via publica",
       LOCOCOR == "5" ~ "Outros",
       TRUE ~ NA_character_
     )
   ) |>
# Unindo a tabela resultante com a tabela grupos_causas pelas colunas com os
#  códigos de causa básica em cada uma
   left_join(grupos_causas, by = c("CAUSA_BASICA_3C" = "CAT"))
```

Agora que já preparamos nosso banco de dados de análise, estamos prontos para realizar o cálculo dos indicadores de mortalidade. Vamos em frente!

3.1 Taxa bruta de mortalidade ou taxa geral de mortalidade

Esse indicador representa o número total de óbitos, por mil (1.000) habitantes, na população residente em determinado local e ano considerado. Ele expressa a **intensidade que a mortalidade atinge uma população**. É um indicador muito usado na demografia, e exige cuidado ao se utilizar para comparações entre populações *distintas*.

$$Taxa\ bruta\ de\ mortalidade = \frac{Número\ total\ de\ óbitos\ de\ residentes}{População\ total\ residente} \times 100mil$$

Para a contagem do número total de óbitos, considere agrupar os dados por ano do óbito. Para isto, utilizaremos a função `count()` para contar o número de óbitos em cada ano.

Vamos guardar os dados resultantes desse comando em um objeto novo: `ob_geral_ac_15_20`, para que possamos utilizá-lo em seguida.

Acompanhe o *script* abaixo e replique em seu `RStudio`:

```
# Criando uma tabela (dataframe) com o nome ob_geral_ac_15_20
  ob_geral_ac_15_20 <- base_obito_ac |>

  # Contando o número de casos por ano de óbito (ANO_OBITO) com o uso
  # da função count()
    count(ANO_OBITO)

# Visualizando a tabela resultante
  ob_geral_ac_15_20
```

```
#>   ANO_OBITO    n
#> 1      2015 3517
#> 2      2016 3763
#> 3      2017 3832
#> 4      2018 4094
#> 5      2019 4098
#> 6      2020 4860
```

A tabela gerada apresentará duas colunas: a `ANO_OBITO`, contendo o ano de ocorrência de cada óbito, e a coluna `n`, que por sua vez contém o total de óbitos registrados em cada um dos anos.

Para o cálculo da taxa bruta de mortalidade, siga o passo a passo abaixo:

1. Utilizando a função `summarise()` vamos resumir os dados de população do Acre, somando as estimativas para cada município e ano.
2. Além disso, vamos estruturar a tabela para ficar na mesma estrutura dos dados de mortalidade. Para isso, vamos utilizar a função `pivot_longer()` do pacote `dplyr` para transformar os dados do formato largo para longo, conforme vimos no Módulo 3. Esta etapa é importante para que criemos as variáveis necessárias para o cálculo da taxa bruta de mortalidade.

Veja o *script* abaixo e escreva em seu `RStudio`:

```
# Criando uma tabela (dataframe) com o nome pop_ac_15_20
  pop_ac_15_20 <- pop_ac |>

  # Calculando o número de casos por ano com a função summarise()
    summarise(
      pop_2015 = sum(`2015`),
      pop_2016 = sum(`2016`),
      pop_2017 = sum(`2017`),
      pop_2018 = sum(`2018`),
      pop_2019 = sum(`2019`),
      pop_2020 = sum(`2020`)
    ) |>

  # Transformando os dados do formato largo para o formato longo
    pivot_longer(cols = pop_2015:pop_2020,
                 values_to = "pop_ac")

# Visualizando a tabela resultante
  pop_ac_15_20
```

```
#> # A tibble: 6 × 2
#>   name     pop_ac
#>   <chr>     <dbl>
#> 1 pop_2015 831665
#> 2 pop_2016 844137
#> 3 pop_2017 856457
#> 4 pop_2018 869265
#> 5 pop_2019 881935
#> 6 pop_2020 894470
```

A tabela resultante irá apresentar em sua primeira coluna (`name`) informações sobre o ano, e na segunda coluna (`pop_ac`) a população estimada para o Acre para cada ano correspondente. Observe que os bancos de dados de óbito e de população possuem a mesma dimensão (2 colunas, 6 linhas) e se referem à mesma unidade de tempo nas linhas: o ano.

3. No terceiro passo utilizaremos a função `bind_cols()` do pacote `dplyr` para unir as tabelas de óbitos e população para calcular a taxa bruta de mortalidade. Vamos executar juntos os códigos do *script* abaixo, e replique-o em seu `RStudio`:

```
# Unindo as duas tabelas com a função bind_cols()
bind_cols(ob_geral_ac_15_20, pop_ac_15_20)
```

```
#>   ANO_OBITO    n     name pop_ac
#> 1      2015 3517 pop_2015 831665
#> 2      2016 3763 pop_2016 844137
#> 3      2017 3832 pop_2017 856457
#> 4      2018 4094 pop_2018 869265
#> 5      2019 4098 pop_2019 881935
#> 6      2020 4860 pop_2020 894470
```

Veja como foi simples unir as duas tabelas utilizando a função `bind_cols()` !

4. Agora podemos criar uma nova coluna contendo a taxa geral de mortalidade usando a função `mutate()`. Vamos lá, acompanhe o *script* abaixo e replique-o no seu computador:

```
# Unindo as duas tabelas com a função bind_cols()
bind_cols(ob_geral_ac_15_20, pop_ac_15_20) |>

  # Criando uma nova coluna com a taxa geral de mortalidade com a função mutate()
  mutate(tx_bruta_ac = (n / pop_ac) * 1000)
```

```
#>   ANO_OBITO    n     name pop_ac tx_bruta_ac
#> 1      2015 3517 pop_2015 831665    4.228866
#> 2      2016 3763 pop_2016 844137    4.457807
#> 3      2017 3832 pop_2017 856457    4.474247
#> 4      2018 4094 pop_2018 869265    4.709726
#> 5      2019 4098 pop_2019 881935    4.646601
#> 6      2020 4860 pop_2020 894470    5.433385
```

Você deverá encontrar esta tabela acima resultante da união de tabelas que foram geradas nos passos anteriores. Esta nova tabela possui a adição de uma coluna nova: a `tx_bruta_ac` que indica a taxa bruta de mortalidade para todo o Estado do Acre, nos anos entre 2015 e 2020. Observe no *output* que a taxa bruta de mortalidade ou taxa geral de mortalidade do Estado do Acre variou entre 4,23 e 5,43 por mil habitantes, nestes anos.

Atenção

Unir tabelas com a função `bind_cols()` do pacote `dplyr` será possível apenas quando ambas as tabelas possuírem a mesma dimensão (número de linhas) e lógica de armazenamento de dados. Nestes casos, os argumentos da função são as tabelas a serem conectadas!!!

3.2 Taxas específicas de mortalidade por idade e sexo

Também podemos calcular se há diferenças entre sexo e idade por uma mesma doença nos óbitos. Para isto, utilizamos a taxa específica de mortalidade por idade e sexo, calculamos o número de óbitos por uma causa de morte específica e a comparamos por 100 mil habitantes residentes em mesmo local e ano determinado.

Observe abaixo, no *script* que selecionamos para análise, os óbitos por causas externas no ano de 2020. Lembre-se que permanecemos utilizando o banco de dados de óbitos do Estado do Acre.

Para a construção do *script* que nos retornará taxas específicas de mortalidade por idade e sexo seguiremos os seguintes passos:

1. filtraremos os seguintes critérios:
    a. registros em que o ano do óbito (`ANO_OBITO`) é igual a 2020;
    b. registros em que as variáveis faixa etária (`FX_ETARIA_11C`) e sexo (`SEXO`) não estão em branco (lembre-se de que o operador `!` tem significado de "NÃO", e irá inverter os Verdadeiro e Falso da operação lógica);
    c. registros cuja causa de morte pertence ao grupo de causas externas de óbito.
2. relizaremos a contagem dos óbitos por faixa etária e sexo;
3. transformaremos a variável `SEXO` de linha para coluna, usando a função `pivot_wider()`;
4. uniremos a tabela de população por sexo e idade {`pop_ac_sexo_idade`}, importada anteriormente;
5. criaremos as variáveis com as taxas usando a função `mutate()`;
6. por fim, selecionaremos as variáveis necessárias para o cálculo da taxa.

Agora, veja com atenção como ficam todos estes passos no código abaixo. Replique a escrita deste *script* em seu `RStudio`:

```
base_obito_ac |>

# Filtrando registros utilizando a função filter()
  filter(

  #Filtrando os dados por óbitos ocorridos em 2020 com a
    ANO_OBITO == 2020,

  # Filtrando os dados e removendo aqueles com valores ausentes para a faixa
  # etária
    !(is.na(FX_ETARIA_11C)),

  # Filtrando os dados e removendo aqueles com valores ausentes para sexo
    !(is.na(SEXO)),

  # Filtrando os dados por óbitos ocorridos por "Causas externas"
    GRUPOS == "Causas externas") |>

# Contando o número de casos por faixa etária e sexo com a função count()
  count(FX_ETARIA_11C, SEXO) |>

# Transformando a tabela do formato longo para o formato largo com a função pivot_wider()
  pivot_wider(names_from = SEXO,
              values_from = n) |>

# Unindo a tabela resultante com a tabela pop_ac_sexo_idade com a função bind_cols()
  bind_cols(pop_ac_sexo_idade) |>

# Criando colunas de taxa específica para homens e mulheres com a função mutate()
  mutate(
    taxa_especifica_homem = masculino / Pop_Masculino * 100000,
    taxa_especifica_mulher = feminino / Pop_Feminino * 100000
  ) |>

# Selecionando apenas as colunas Faixa Etaria 1, taxa_especifica_homem e
# taxa_especifica_mulher com a função select()
  select("Faixa Etaria 1",
         taxa_especifica_homem,
         taxa_especifica_mulher)
```

```
#> # A tibble: 11 × 3
#>    `Faixa Etaria 1` taxa_especifica_homem taxa_especifica_mulher
#>    <chr>                            <dbl>                  <dbl>
#>  1 0 a 4 anos                        11.8                   2.47
#>  2 5 a 9 anos                        11.5                   7.23
#>  3 10 a 14 anos                      22.0                   6.83
#>  4 15 a 19 anos                     179.                   35.1
#>  5 20 a 29 anos                     177.                   22.0
#>  6 30 a 39 anos                     172.                   18.9
#>  7 40 a 49 anos                     121.                   16.9
#>  8 50 a 59 anos                     110.                   14.4
#>  9 60 a 69 anos                      94.7                  14.3
#> 10 70 a 79 anos                     111.                   18.1
#> 11 80 anos e mais                   169.                   39.4
```

Observe no *output* que, para o ano de 2020 no Estado do Acre, a taxa de mortalidade por causas externas em homens é muito superior à taxa de mortalidade por causas externas em mulheres. A faixa etária mais acometida é a de 15 a 19 anos em homens e 80 anos e mais em mulheres.

3.3 Mortalidade proporcional por grupo de causas

Um importante indicador é o de mortalidade proporcional. Ele representa a distribuição percentual (`%`) de óbitos e pode ser analisada com foco em grupos de causas de óbito. Lembre-se que sempre deverá analisar uma mesma população residente em um mesmo espaço geográfico (local) e no tempo (dias, meses ou anos).

Este indicador é importante para medir **a carga de cada grupo de causa no total de óbitos da região geográfica analisada**. Pode ser calculado matematicamente da seguinte forma:

$$\textit{Mortalidade proporcional por grupos de causas} = \frac{\textit{Números de óbitos de residentes por grupos de causas}}{\textit{Número total de óbitos de residentes}} \times 100$$

Para todo cálculo que representar uma proporção pode-se utilizar cálculos simples para a porcentagem, ou seja, multiplicado por 100.

Para o cálculo deste indicador no `R`, você deverá criar uma tabela com frequência absoluta e frequência relativa, cuja coluna de frequência relativa representará o resultado do cálculo da mortalidade proporcional. Para esta etapa, siga o passo a passo abaixo:

1. Primeiro, selecionaremos apenas os dados por óbitos ocorridos em 2020 com a função `filter()`.
2. Depois iremos realizar o cálculo da frequência absoluta de óbitos por grupos com a função `count()`.
3. E por fim, utilizamos a função `mutate()` para calcular a frequência relativa de óbitos para cada grupo.

Agora, observe o *script* abaixo e replique em seu `RStudio`.

```
base_obito_ac |>

# Filtrando os dados por óbitos ocorridos em 2020 com a função filter()
  filter(ANO_OBITO == 2020) |>

# Contando o número de casos por grupos com a função count()
  count(GRUPOS) |>

# Calculando o número de mortes proporcionais em cada grupo pelo número
#  total com a função mutate()
  mutate(mort_proporcional = (n / sum(n)) * 100)
```

```
#>                                          GRUPOS    n mort_proporcional
#> 1 Algumas afecções originadas no período perinatal  125         2.5720165
#> 2                                Causas externas  596        12.2633745
#> 3                           Causas Mal Definidas  662        13.6213992
#> 4                                Causas Maternas    5         0.1028807
#> 5             Doenças do aparelho circulatório  771        15.8641975
#> 6             Doenças do aparelho respiratório  488        10.0411523
#> 7          Doenças infecciosas e parasitárias 1006        20.6995885
#> 8                                  Neoplasias  539        11.0905350
#> 9                               Outros grupos  668        13.7448560
```

Assim, ao observar o *output* do código executado podemos concluir que o maior grupo de causa de morte no Acre é o de Doenças infecciosas e parasitárias, com 20,69% de todas as mortes ocorridas no Estado. Essa informação é muito importante para a vigilância, pois o serviço precisa conhecer as principais causas de mortalidade e desencadear ações de saúde e intersetoriais.

3.4 Taxa de mortalidade infantil

A taxa de mortalidade infantil representa o número de óbitos ocorridos em crianças menores de um ano de idade por mil nascidos vivos no mesmo período. Este indicador é muito importante por permitir avaliar as condições de vida e de saúde de uma dada população. Com o cálculo da sua taxa, é possível estimar o risco de uma criança morrer antes de chegar a um ano de vida, sendo consideradas como altas (50 ou mais), médias (20-49) e baixas (menos de 20).

As taxas de mortalidade infantil, em especial a mortalidade pós-neonatal, quando **elevadas refletem as precárias condições de vida e saúde e valores abaixo do nível de desenvolvimento social e econômico**. Já taxas reduzidas podem sinalizar bons indicadores sanitários e sociais, mas também podem encobrir más condições de vida em segmentos sociais específicos.

No caso do cálculo da mortalidade infantil e de seus componentes, o denominador é o número de nascidos vivos de mães residentes do local. Observe como é possível calcular este indicador matematicamente pela seguinte expressão:

$$Taxa\ de\ Mortalidade\ Infantil = \frac{Número\ de\ óbitos\ de\ residentes\ menores\ de\ um\ ano\ de\ idade}{Número\ de\ nascidos\ vivos\ de\ mães\ residentes} \times 1000$$

Agora, vamos escrever o script que corresponderá à fórmula matemática e que permite que seu cálculo possa ser realizado de maneira automatizada. Observe abaixo e replique no seu `RStudio`:

```
base_obito_ac |>

# Filtrando os dados por registros de casos com menos de 1 ano com a função
# filter()
  filter(FX_ETARIA_M1ANO == "< 1 ano") |>

# Contando o número de casos por ano de óbito com a função count()
  count(ANO_OBITO) |>

# Unindo a tabela resultante com a tabela nv_ac com a função bind_cols()
  bind_cols(nv_ac) |>

# Calculando a taxa de mortalidade infantil com a função mutate()
  mutate(tx_mortalidade_infantil = (n / n_nascidos_vivos) * 1000)
```

```
#>   ANO_OBITO   n Ano do nascimento n_nascidos_vivos tx_mortalidade_infantil
#> 1      2015 291              2015            16980                17.13781
#> 2      2016 239              2016            15773                15.15248
#> 3      2017 223              2017            16358                13.63247
#> 4      2018 273              2018            16543                16.50245
#> 5      2019 259              2019            16280                15.90909
#> 6      2020 243              2020            15142                16.04808
```

A taxa de mortalidade infantil por mil nascidos vivos no Acre chegou a diminuir de 17,13 em 2015 para 13,63 em 2017, mas voltou a aumentar e chegou a 16,04 em 2020.

3.5 Taxa de mortalidade materna

A taxa de mortalidade materna é calculada a partir do número de óbitos maternos ocorridos a cada 100 mil nascidos vivos de mães residentes. Esta taxa é **definida como a morte durante a gravidez ou no prazo de 42 dias após o final da gestação**. A taxa de mortalidade materna é um indicador muito utilizado para entender e comparar um importante problema de saúde pública global: **a condições de saúde das mulheres, o desenvolvimento econômico e as desigualdades sociais** dos países.

Observe abaixo como é possível calcular a taxa de mortalidade materna matematicamente:

$$Taxa\ de\ Mortalidade\ Materna = \frac{\begin{array}{c}Número\ de\ óbitos\ de\ mulheres\ residentes \\ (por\ causas\ de\ morte\ materna)\end{array}}{\begin{array}{c}Número\ de\ nascidos \\ vivos\ de\ mães\ residentes\end{array}} \times 100\ mil$$

Agora, vamos escrever o *script* que corresponderá à fórmula matemática e permitirá que seu cálculo possa ser realizado de maneira automatizada, replique o código abaixo em seu `RStudio`:

```
base_obito_ac |>

# Utilizando a função filter() para diversos critérios de filtragem de dados
  filter(

  # Filtrando os dados por registros de casos do sexo feminino
    SEXO == "feminino",

  # Filtrando os registros com casos nas faixas etárias entre 10 e 49 anos
    FX_ETARIA_11C %in% c("10 a 14", "15 a 19",
                         "20 a 29", "30 a 39",
                         "40 a 49"),

  # Filtrando os registros de óbitos por causas maternas
    GRUPOS == "Causas Maternas"
  ) |>

# Contando o número de casos por ano de óbito com a função count()
  count(ANO_OBITO) |>

# Unindo a tabela resultante com a tabela nv_ac com a função bind_cols()
  bind_cols(nv_ac) |>

# Calculando a taxa de mortalidade materna com a função mutate()
  mutate(razao_mort_materna = (n / n_nascidos_vivos) * 100000)
```

```
#>   ANO_OBITO  n Ano do nascimento n_nascidos_vivos razao_mort_materna
#> 1      2015 11              2015            16980           64.78210
#> 2      2016  9              2016            15773           57.05953
#> 3      2017  8              2017            16358           48.90573
#> 4      2018  9              2018            16543           54.40368
#> 5      2019  8              2019            16280           49.14005
#> 6      2020  5              2020            15142           33.02074
```

Observe que o nosso *output* informa para cada ano o total de óbitos maternos (coluna `n`), o número de nascidos vivos (coluna `n_nascidos_vivos`), e o valor da razão de mortalidade materna (coluna `razao_mort_materna`).

Ao analisarmos a tabela construída, é possível observar melhora na taxa de mortalidade materna ao longo desses 6 anos, o que pode indicar esforços do Estado do Acre para interferir sobre o indicador. Lembre-se dos vieses que influenciam este indicador como a subnotificação do denominador (número de nascidos vivos) e problemas de investigação do óbito materno (número de óbitos por causas de morte materna).

Para tornar sua análise ainda mais aprofundada sugere-se que você análise além da causa específica da morte materna, também a idade da mulher ao morrer, o estado marital, o número de filhos tidos e os ainda vivos, a paridade, o intervalo entre gestações, consultas no pré-natal, hábito de fumar, de beber e outras. Algumas destas variáveis, não são obtidas a partir dos atestados de óbito, necessitando análises do SINASC, e-SUS AB PEC ou outros bancos de dados disponíveis em seu município.

Lembre-se que todo e qualquer indicador de saúde deve prezar pela simplicidade para que as informações sejam transmitidas de forma clara e sem inconsistências.

Caso não tenha compreendido os dados ou valores, retorne ao seu cálculo e refaça. Torne-o simples!

Para consultar mais indicadores de saúde, visite o site das fichas de qualificação de indicadores da Ripsa.

http://fichas.ripsa.org.br/2012/.

Com base nos códigos até agora descritos você já pode expandir suas análises para diferentes doenças e agravos, locais, anos, faixas etárias, etc. Mas vamos adiante. Há mais possibilidades à frente.

4. Análises de variáveis categóricas

Na análise exploratória em saúde, quando avaliamos duas variáveis categóricas, ou seja, variáveis qualitativas categorizadas em classes (por exemplo: variável sexo, categoria: feminino e masculino), o resumo dos dados se dá por meio da contagem de indivíduos que pertencem às categorias de ambas as variáveis, simultaneamente.

Para isto, costumamos montar tabelas de análises chamadas de **tabelas 2x2** ou **tabelas de contingência**. Muito utilizadas quando calculamos **taxa de ataque**, uma espécie de coeficiente ou taxa de incidência que analisa uma determinada doença comparando um grupo de pessoas expostas ao mesmo risco limitadas a uma área bem definida. O cálculo da **taxa de ataque** é muito útil para investigar e analisar surtos de doenças ou agravos à saúde em locais delimitados.

Na representação abaixo utilizaremos um exemplo em que iremos verificar se um grupo de pessoas em um município possui maior chance de adoecer e após ter frequentado um restaurante com suspeita de contaminação nos alimentos. Desta forma, temos duas variáveis ("estado de saúde" e "restaurante"), e duas categorias em cada uma (saudável/doente, e frequentou/não frequentou, respectivamente). Em cada célula, temos as contagens de indivíduos que pertencem às respectivas categorias das variáveis cruzadas (a, b, c, d) :

Figura 3: Tabela de contingência (tabela 2x2) de exemplo.

	Estado de Saúde		
Restaurante	**Saudável**	**Doente**	**Total**
Não frequentou	a	b	a + b
Frequentou	c	d	c + d
Total	a + c	b + d	a + b + c + d

Convencionalmente, as colunas da tabela 2x2 representam a presença ou ausência de um evento ou doença e as linhas a presença ou ausência de exposição a este evento. Da seguinte forma:

- A célula "a" representa os indivíduos que permaneceram saudáveis e não frequentaram o restaurante suspeito.
- A célula "b" representa indivíduos que adoeceram e não frequentaram o restaurante suspeito.
- A célula "c" representa os indivíduos saudáveis e que frequentaram o restaurante suspeito.
- A célula "d" representa os indivíduos que adoeceram e frequentaram o restaurante suspeito.
- Nas marginais da tabela, há os totais, somando linhas e colunas.

Agora, vamos acompanhar algumas análises em que as tabelas 2x2 serão úteis na rotina de trabalho da vigilância em saúde. Vamos lá!

4.1 Associações entre variáveis (Teste qui-quadrado)

Algumas vezes, durante a rotina de análise, precisamos saber se a frequência de uma determinada variável muda quando outra variável está presente e, assim, saber se alguma evidência de associação entre elas ou seria somente uma relação ao acaso. Para isso, podemos utilizar a tabela de contingência (tabela 2x2) e o teste estatístico qui-quadrado.

Definimos a existência de uma associação quando a probabilidade de ocorrência de um evento ou doença se altera quando alguma característica ou outro evento está presente.

Isto acontece com frequência na vigilância, quando estudamos a ocorrência de doença em determinados locais ou testamos hipóteses a partir da investigação de um surto. Dessa forma, estudamos se há possível associação entre uma exposição (fator de risco) e um efeito (doença).

Vamos lá, agora é hora de praticar com exemplos!

Considere que o profissional de vigilância em saúde necessita verificar se a classificação final de dengue no Estado de Rosas possui associação com alguma característica regional como, por exemplo, o local onde o paciente mora. Ou seja, o profissional deve investigar **se a frequência de casos prováveis de dengue é diferente entre os municípios de residência dos casos**.

Essa investigação é oportuna quando há suspeita de que municípios localizados mais próximos à Capital, por exemplo, poderiam ter mais acesso a estruturas melhores de saúde. Por outro lado, municípios localizados muito distantes das regiões centrais, podem apresentar desigualdades que impactam em muitos aspectos da saúde púbica, inclusive, nas ações contra à dengue. Perceba que essa mesma comparação poderia ser facilmente feita intramunicipal, verificando a diferença entre bairros mais próximos e mais afastados do centro.

Para investigar essa situação, vamos utilizar o agravo dengue da base de notificação do Estado de Rosas. Acompanhe o passo a passo a seguir:

1. Utilize o banco de dados {`NINDINET.dbf`} importado anteriormente;
2. Filtre os registre para os casos notificados de Dengue;
3. Categorize o município de residência (`ID_MN_RESI`) considerando o seguinte:
    - Registros de casos que residem no município Prímula do Estado de Rosas (código 610213) serão categorizados como da região da "Capital";
    - Registros de casos que residem nos demais municípios do Estado de Rosas serão categorizados como da região do "Interior".
4. Categorize a classificação final de dengue (`CLASSI_FIN`) da seguinte forma:
    - Registros de casos notificados como "1", "2", "3", "4" e "8", serão categorizados como "**Caso provável**";
    - Registros de casos notificados como "5", serão categorizados como "**Caso descartado**".

Acompanhe e replique o *script* abaixo em seu `RStudio`, que demonstra as etapas de 1 a 4, indicadas anteriormente:

```
# Criando um novo dataframe com a tabela {`Dados`}
dados_classificados <- dados |>

  # Filtrando os agravos de dengue (código "A90") com a função filter()
  filter(ID_AGRAVO == "A90") |>

  # Criando novas colunas com a função mutate()
  mutate(

    # Classificando o município de residência do paciente em Prímula (código 610213)
    # como "Capital" e os demais como "Interior" com uso da função if_else().
    # Ou seja, se o município for igual a 610213 será classificado como Capital e, se não, será
    # classificado como Interior. Caso houver registros em branco (NA),
    #estes continuarão como (NA)
    regiao = if_else(ID_MN_RESI == "610213", "Capital", "Interior",
NA_character_),

    # Classificando os casos em diagnósticos de "casos prováveis" e "descartado"
    # com a função case_when()
    classificacao = case_when(
      CLASSI_FIN %in% c("1", "2", "3", "4", "8") ~ "Casos prováveis",
      CLASSI_FIN == "5" ~ "Casos descartados",
      TRUE ~ NA_character_
    )
  )
```

5. No quinto passo, vamos cruzar os dados em que realizamos a classificação da região de residência e do diagnóstico de dengue utilizando a função `tbl_cross()` do pacote `gtsummary`. Para isso, Iremos utilizar os seguintes argumentos:

   - `row`: indicação de uma variável categórica na linha;
   - `col`: indicação de uma variável categórica na coluna;
   - `percent`: definição se a porcentagem será calculada por colunas ou linhas (*column* ou *row*);
   - `missing:` definição se os valores em branco serão incluídos na tabela.

Agora veja como o passo 5 pode ser escrito em um *script* abaixo e replique em seu `RStudio`:

```
# Criando uma tabela no formato do pacote gtsummary com o nome tabela
tabela_resid_diag_dengue <- dados_classificados |>

  # Criando uma tabela de resumo do cruzamento de informações sobre região e tipo de diagnóstico
  tbl_cross(
    row = regiao,
    col = classificacao,
    percent = "row",
    missing = "no"
  )
```

Agora, vamos visualizar a tabela criada, para isso digite apenas seu nome {`tabela_resid_diag_dengue`} e clique no botão "*Run*". Fique atento pois a tabela será mostrada no painel *`Viewer`* do `RStudio` e não no Painel `Console`!

Vamos lá, acompanhe o script, replique no seu `RStudio`:

```
# visualizando a tabela "tabela"
tabela_resid_diag_dengue
```

	classificacao		
	Casos descartados	Casos prováveis	Total
regiao			
Capital	442 (3.6%)	11,672 (96%)	12,114 (100%)
Interior	13 (4.0%)	314 (96%)	327 (100%)
Total	455 (3.7%)	11,986 (96%)	12,441 (100%)

Atenção

Você pode ter observado que o `R` emiteu um aviso informando que as observações em branco não foram incluídas na análise. Este tipo de alerta é muito importante para nos apoiar na interpretação do que está sendo analisado.

Atente-se sempre às mensagens no output do `R`!

Observe que tanto a região da Capital quanto o do Interior do Estado de Rosas possuem porcentagens semelhantes de casos prováveis e de casos descartados. Avaliando a última linha da tabela gerada no *output* do código, **perceberemos que independente da região, 96% dos casos notificados são casos prováveis**.

Podemos pensar que, se as variáveis cruzadas não fossem relacionadas, observaríamos as porcentagens muito próximas em cada região. Isso de fato acontece! Portanto, parece não haver associação entre elas. Veja como é importante este relacionamento.

Agora, vamos explorar a presença de significância estatística aplicando o teste *qui-quadrado*.

Para construir o teste *qui-quadrado* precisamos admitir:

1. **A hipótese de que a região de residência não tem associação com a classificação final do caso** (nesse caso será nossa hipótese nula, como é chamada em estatística, e será a hipótese que vamos testar).
2. A hipótese de que as variáveis possuem alguma associação será definida como alternativa;
3. Vamos definir que o nível de significância estatística deve ser menor que 0,05 para rejeitarmos a hipótese nula, ou seja, de que **não existe associação da região de residência com a classificação final**.

Pronto, agora vamos realizar o teste qui-quadrado para verificar se esta associação é somente devido ao acaso. Podemos fazer isso através da função `add_p()` do pacote `gtsummary`. Ao adicionarmos esta função, as células da tabela são verificadas e o teste estatístico adequado é realizado.

Acompanhe o *script* abaixo e digite estes comandos em seu `RStudio`:

```
# Adicionado o teste estatítico com a função add_p()
tabela_resid_diag_dengue |> add_p()
```

	classificacao			
	Casos descartados	Casos prováveis	Total	p-value[1]
regiao				0.8
Capital	442 (3.6%)	11,672 (96%)	12,114 (100%)	
Interior	13 (4.0%)	314 (96%)	327 (100%)	
Total	455 (3.7%)	11,986 (96%)	12,441 (100%)	

[1] Pearson's Chi-squared test

Observe que o resultado do valor do `p` foi 0,8. O que isso significa? Como 0,8 é maior que 0,05, não podemos rejeitar a hipótese nula que definimos anteriormente. Assim, **temos evidência que não há associação entre a região de residência e a classificação final de Dengue**.

Interessante, não é mesmo? Isso faz sentido, pois a dengue está muito espalhada pelo território, sem distinção de local de residência. O mosquito *Aedes aegypti* não reconhece as barreiras ou fronteiras geográficas.

O teste *qui-quadrado* χ^2 também pode ser executado utilizando a função `chisq.test()`, nativa da linguagem `R`. O argumento essencial dessa função é um *dataframe* contendo os dados da tabela 2x2. Observe o *script* abaixo, no qual criamos o *dataframe* e usamos o *pipe* (`|>`) para executarmos diretamente a função `chisq.test`.

A tabela foi montada conforme o *script* abaixo. Digite os comandos a seguir em seu `RStudio`:

```
#Inserindo os dados para construir a tabela (dataframe) 2x2
#Capital = c(casos prováveis, casos descartados),
#Interior = c(casos prováveis, casos descartados)
data.frame(Capital = c(11672, 442),
           Interior = c(314, 13)) |>
  #calculando agora o teste qui-quadrado
  chisq.test()
```

```
#>
#>  Pearson's Chi-squared test with Yates' continuity
correction
#>
#> data:  data.frame(Capital = c(11672, 442), Interior =
c(314, 13))
#> X-squared = 0.026064, df = 1, p-value = 0.8717
```

Percebeu que o valor do `p` é bem próximo? O pacote `gtsummary` utliza essa mesma função para realizar os cálculos. Escolha o que achar mais fácil de usar.

Atenção

O teste *qui-quadrado* possui limitações nos cálculos quando o total de pessoas a serem avaliadas é pequeno.

Não há uma definição clara do número, mas valores menores que 30 pessoas ou células da tabela 2x2 com menos de 5 observações são indicativos que um teste estatístico mais apropriado deve ser escolhido, por exemplo, o **teste exato de Fisher**.

Vamos praticar! Imagine agora que temos apenas 3 casos prováveis na Capital e 2 descartados, e no Interior 2 casos prováveis e 3 descartados, criaremos um data.frame com a estrutura de tabela 2x2 usando a função `fisher.test()` do `R`. Iremos considerar as seguintes hipoteses:

`Ho`: proporção de casos prováveis na "Capital" é igual a proporção de casos prováveis na "Interior".

`H1`: proporção de casos prováveis na "Capital" é maior que a proporção de casos prováveis na "Interior".

Acompanhe o script abaixo e reproduza em seu `RStudio`:

```
#Inserindo os dados para construir a tabela (dataframe) 2x2
data.frame(Capital = c(3, 2),
           Interior = c(2, 3)) |>

# O teste exato de Fisher é feito usando a função `fisher.test()`
fisher.test(dataframe,
          # argumento `alternative`indica a hipótese alternativa do teste
            alternative = "greater",
        # argumento `conf.int` indica que o intervalo de confiança deve ser construído
            conf.int = TRUE,
      # argumento `conf.level` indica o nível de confiança a ser utilizado para a construção do intervalo
            conf.level= 0.95
)
```

```
#>
#>  Fisher's Exact Test for Count Data
#>
#> data:  data.frame(Capital = c(3, 2), Interior = c(2,
3))
#> p-value = 0.5
#> alternative hypothesis: true odds ratio is greater
than 1
#> 95 percent confidence interval:
#>  0.1541449       Inf
#> sample estimates:
#> odds ratio
#>   2.069959
```

Observe que no *output* serão apresentados o seguintes retornos:

- p-valor do teste,
- a hipótese alternativa em consideração,
- o intervalo de confiança construído baseado na hipótese alternativa, e
- a estimativa da razão de chances com base na tabela de contingência.

Com o teste podemos concluir que não há evidências para rejeitar Ho (p=0,50), ou seja, não há evidências de que na Capital os casos prováveis é maior que o número de casos provavéis no população do Interior.

Para o aprofundamento no tema, livros de bioestatística devem ser consultados.

5. Medidas de associação

Sempre quando estiver realizando uma análise epidemiológica e a associação entre variáveis estiver presente é possível quantificar a magnitude da diferença de chance ou de risco. Para isto, em epidemiologia, usamos duas medidas para nos apoiar nesta análise:

- O *Risco Relativo* (RR) e
- O *Odds Ratio* (OR).

Nas subseções logo abaixo iremos rever os conceitos e aprender a aplicá-los no dia a dia, dando robustez às análises realizadas e qualificando ainda mais a rotina de análise dos serviços de vigilância em saúde municipais.

5.1 Risco Relativo - RR

O *Risco Relativo* é uma razão de incidências que estima a magnitude das diferenças de incidência entre os grupos expostos e não expostos a um fator e a ocorrência da doença.

Em poucas palavras, calcular o valor do RR expressa **quantas vezes é maior (ou menor) o RISCO de desenvolver a doença no grupo que foi exposto a um determinado fator em comparação a um grupo que não foi exposto**. Expressa uma comparação matemática da incidência entre os grupos. Veja seu cálculo matemático:

$$Risco\ Relativo\ (RR) = \frac{Incidência\ no\ grupo\ exposto}{Incidência\ no\ grupo\ não\ exposto}$$

- Quando **o resultado dessa comparação é igual a 1**: indica que a incidência da doença no grupo exposto é igual à do grupo não exposto, portanto não há associação entre exposição e doença.
- Quando **o resultado é maior que 1**: indica associação positiva ou risco aumentado entre os expostos ao fator estudado (potencial fator de risco).
- Quando **o resultado é menor que 1**: indica um risco diminuído entre os expostos ao fator estudado (potencial fator de proteção).

Para analisar esta situação, observe a tabela abaixo que faz uma simulação de grupos expostos e não expostos (na linha) e se houve ou não a doença (na coluna). As letras expressam as seguintes interseções:

- Pessoa que foi exposta e ficou doente: letra "`a`";
- Pessoa que foi foi exposta e **não** ficou doente: letra "`b`";
- Pessoa que **não** foi exposta e ficou doente: letra "`c`";
- Pessoa que **não** foi exposta e **não** ficou doente: letra "`d`".

Logo, a incidência da doença em quem foi exposto é representada por `a/a+b` e a incidência da doença em quem não foi exposto é representada por `c/c+d`.

Figura 4: Tabela 2x2 para avaliação do Risco Relativo.

Grupo	Doente	Não doente	Total	Incidência
Exposto	a	b	a + b	a/a+b
Não exposto	c	d	c + d	c/c+d

A partir da tabela acima o cálculo do Risco Relativo seria o seguinte:

$$Risco\ Relativo\ (RR) = \frac{Incidência\ no\ grupo\ exposto}{Incidência\ no\ grupo\ não\ exposto} = \frac{\frac{a}{a+b}}{\frac{c}{c+d}}$$

Agora, vamos praticar no `R` utilizando um exemplo comum na rotina de vigilância em saúde!

Considere que você foi chamado para participar da investigação de um surto de gastroenterite (DTA) entre pessoas que participaram de uma festa de casamento. Ao total, 80 pessoas estiveram presentes no jantar que ocorreu em uma igreja no dia 18 de abril. O jantar foi realizado das 18 às 23 horas. Durante este período, os alimentos ficaram expostos em uma mesa e disponíveis aos convidados.

A equipe da Vigilância Epidemiológica de Rosas conseguiu entrevistar 75 pessoas, sendo coletados os seguintes dados:

- alimentos que foram consumidos pelas pessoas;
- se a pessoa ficou doente (considerando a definição de caso prevista para o surto);
- a hora do jantar;
- o dia e hora do início dos sintomas;
- a idade e o sexo das pessoas.

Os dados coletados foram tabulados e enviados ao seu e-mail no formato texto (`.csv`). Agora você deverá seguir o passo a passo abaixo para analisá-los:

1. Importaremos o banco de dados {`base_surto.csv`}, localizada no menu lateral "Arquivos", do módulo. Agora iremos analisá-lo com auxílio do `R`, replique o script abaixo em seu `RStudio`:

```
# Carregando o banco de dados do surto e criando uma tabela (dataframe) com o nome base_surto
base_surto_18_04 <- read_csv("Dados/base_surto.csv")
```

```
#> Rows: 75 Columns: 21
#> ── Column specification ────────────────────────────────────────────────────
#> Delimiter: ","
#> chr  (17): sexo, doente, data_inicio_sintomas, presunto, espinafre, pure_bat...
#> dbl   (2): id, idade
#> time  (2): hora_jantar, hora_inicio_sintomas
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Como já deve ser de seu conhecimento, a investigação de surto pela vigilância visa constatar a causa do surto, suas características quanto ao local, pessoa e tempo e quais ações necessárias para seu término. Considere também que a análise pode determinar se existe alguma associação entre consumo de algum alimento e a ocorrência da doença. Portanto, vamos construir as tabelas 2x2 para cada um dos supostos fatores de risco.

2. Como recurso estético, vamos deixar a tabela um pouco mais compacta, executando a função `theme_gtsummary_compact()`, considerando que a tabela pode ficar grande e de difícil visualização.
3. Vamos retirar as variáveis que não vamos utilizar neste momento utilizando a função `select()`. Note que estamos utilizando o símbolo de menos (`-`) para retirar as variáveis de identificação (`id`), data e hora de início de sintomas e hora do jantar.
4. Em seguida, utilizaremos a função `tbl_summary()` do pacote `gtsummary` apresentada anteriormente. Faremos um cruzamento entre as variáveis sobre alimentos ingeridos e a variável com status da doença.
5. E como último passo aplicaremos a função `add_p()` para o cálculo do teste estatístico adequado para verificar se há associação entre os fatores e o status dos doentes.

Executaremos agora juntos os passos 2 a 5, escrevendo-os no *script* abaixo:

```
# Utilizando a função theme_gtsummary_compact() para facilitar a visualização da tabela que será gerada
theme_gtsummary_compact()

base_surto_18_04 |>

  # Removendo as colunas id, data_inicio_sintomas, hora_inicio_sintomas e hora_jantar com a função select()
  select(-id,
         -data_inicio_sintomas,
         -hora_inicio_sintomas,
         -hora_jantar) |>

  # Criando uma tabela com resumo das informações de acordo com a coluna "doente"
  tbl_summary(by = doente, missing = "no") |>

  # Adicionado o p-valor com a função add_p()
  add_p()
```

Characteristic	Nao, N = 29[1]	Sim, N = 46[1]	p-value[2]
idade	35 (14, 50)	38 (17, 59)	0.3
sexo			0.15
F	14 (48%)	30 (65%)	
M	15 (52%)	16 (35%)	
presunto			0.7
Nao	12 (41%)	17 (37%)	
Sim	17 (59%)	29 (63%)	
espinafre			0.9
Nao	12 (41%)	20 (43%)	
Sim	17 (59%)	26 (57%)	
pure_batata			>0.9
Nao	14 (50%)	23 (50%)	
Sim	14 (50%)	23 (50%)	
salada_repolho			0.7
Nao	19 (66%)	28 (61%)	
Sim	10 (34%)	18 (39%)	
gelatina			0.3
Nao	22 (76%)	30 (65%)	
Sim	7 (24%)	16 (35%)	
paozinho			0.4
Nao	13 (45%)	25 (54%)	
Sim	16 (55%)	21 (46%)	
pao_centeio			0.5
Nao	20 (69%)	28 (61%)	
Sim	9 (31%)	18 (39%)	
leite			0.6
Nao	27 (93%)	44 (96%)	
Sim	2 (6.9%)	2 (4.3%)	
cafe			>0.9
Nao	17 (59%)	27 (59%)	
Sim	12 (41%)	19 (41%)	
agua			0.4
Nao	18 (62%)	33 (72%)	
Sim	11 (38%)	13 (28%)	
bolo			0.2
Nao	16 (55%)	19 (41%)	
Sim	13 (45%)	27 (59%)	
sorvete_creme			<0.001
Nao	18 (62%)	3 (6.5%)	
Sim	11 (38%)	43 (93%)	
sorvete_chocolate			0.076
Nao	7 (24%)	20 (44%)	
Sim	22 (76%)	25 (56%)	
salada_frutas			>0.9
Nao	27 (93%)	42 (91%)	
Sim	2 (6.9%)	4 (8.7%)	

[1] Median (IQR); n (%)

[2] Wilcoxon rank sum test; Pearson's Chi-squared test; Fisher's exact test

Pronto, verifique se os seguintes resultados para os que ficaram doente foi o seguinte:

- 65% eram mulheres e com 38 anos de idade mediana.
- Alguns alimentos não foram muito consumidos por este grupo como salada de repolho (61%), pão de centeio (61%), gelatina (65%), água (72%), salada de frutas (91%) e leite (96%).
- Por outro lado, a maioria das pessoas que consumiram o sorvete de creme desenvolveram gastroenterite (93%).

Nessa situação, perceba que o sorvete de creme é o único alimento que as pessoas foram "expostas" e que apresentou valor de `p` menor que 0,05 no teste estatístico de associação. **Temos, portanto, nosso principal suspeito**. Perceba também que, agora, será preciso comparar quem foi exposto ou não ao fator de risco (sorvete de creme) e que desenvolveram a doença (gastroenterite) ou não. Vamos calcular o Risco Relativo (RR)!

6. Para calcular o RR, vamos precisar do pacote `epitools`, muito útil para cálculo de medidas de associação e já carregado no início do módulo.
7. Para o cálculo do Risco Relativo vamos utilizar a função `riskratio()` do pacote `epitools`. Essa função possui três argumentos principais:
    - `x`: vetor contendo os números da tabela 2x2;
    - `method`: método para calcular o Risco Relativo. Usaremos o método de Wald;
    - `rev`: ordem das linhas e colunas que serão realizados os cálculos. Usaremos "both".

O argumento `conf.level` define o nível de confiança das estimativas. Por padrão ele já é preenchido como 0,95 (95%). Assim, não é necessário inseri-lo na função. Caso queira alterar, basta definir na função com o valor correspondente.

Vejamos como implementamos o cálculo do RR para o sorvete de creme. Vamos inserir o vetor com os dados do sorvete de creme na função `riskratio()`, com os argumentos citados acima. Vamos salvar o resultado em um objeto chamado `resultado_rr`.

Acompanhe o *script* abaixo e replique-o em seu `RStudio`:

```
# Calculando o risco relativo com a função riskratio() e criando um objeto do
# com o nome resultado_rr_surto
resultado_rr_surto <- riskratio(x = c(43, 11, 3, 18),
                                method = "wald",
                                rev = "both")
```

8. O objeto `resultado_rr` que retorna da função é um objeto do tipo lista composta por quatro itens: os dados (data), o cálculo do risco relativo com intervalo de confiança (*measure*), o `p` valor (*p.value*) e correção (*correction*), ou seja, se houve correção estatística. Você os estudará de forma detalhada mais a adiante. Agora, escreva os comandos abaixo em seu `RStudio`:

```
# Visualizando os nomes do objeto resultado_rr_surto
resultado_rr_surto
```

```
#> $data
#>           Outcome
#> Predictor  Disease2 Disease1 Total
#>   Exposed2       18        3    21
#>   Exposed1       11       43    54
#>   Total          29       46    75
#>
#> $measure
#>           risk ratio with 95% C.I.
#> Predictor  estimate   lower    upper
#>   Exposed2 1.000000      NA       NA
#>   Exposed1 5.574074 1.93834 16.02934
#>
#> $p.value
#>           two-sided
#> Predictor    midp.exact fisher.exact   chi.square
#>   Exposed2           NA           NA           NA
#>   Exposed1 2.698215e-07 2.597451e-07 1.813314e-07
#>
#> $correction
#> [1] FALSE
#>
#> attr(,"method")
#> [1] "Unconditional MLE & normal approximation (Wald) CI"
```

Observe que os itens principais, visualizados, são:

```
-   `data`: contém a matriz com os dados utilizados.
-   `measure`: matriz com o resultado dos cálculos do risco relativo
com intervalo de confiança.
-   `p.value`: matriz com resultados dos valores de p dos testes
estatísticos realizados.
```

9. Como vamos precisar apenas dos cálculos do Risco Relativo (RR), vamos utilizar somente o segundo elemento da lista, chamado "*measure*". Veja no *script* abaixo como o acessamos e replique-o em seu computador:

```
# Selecionando o objeto "measure" da lista resultado_rr_surto
resultado_rr_surto[["measure"]]
```

```
#>           risk ratio with 95% C.I.
#> Predictor  estimate   lower    upper
#>   Exposed2 1.000000      NA       NA
#>   Exposed1 5.574074 1.93834 16.02934
```

Agora podemos interpretá-lo. Considerando os padrões da tabela 2x2, a primeira linha foi a referência considerada para o cálculo dos expostos.

Assim, **uma forte associação entre consumo de sorvete de creme e gastroenterite foi constatado**, pois o valor do RR é maior que 1. Ou seja, **pessoas que consumiram sorvete de creme apresentaram 5,57 vezes o risco de desenvolver gastroenterite, em comparação a quem não consumiu o sorvete**.

Ufa! Conseguimos calcular o RR com sucesso. Agora vamos aprender a calcular o *Odds Ratio*!!

5.2 Odds ratio - OR

Odds ratio (OR) é uma razão que estima **quantas vezes é maior (ou menor) a CHANCE de exposição a um determinado fator no grupo que já manifestou a doença em relação a um grupo não doente**. Ou seja, odds ratio identifica a associação entre a exposição e doença entre pessoas que já desenvolveram a doença.

O termo *odds* se refere à probabilidade em português, mas alguns autores traduzem como "chance" e tem como sinônimos "razão de chances", "razão de produtos cruzados" ou simplesmente "OR", quando há a divisão das chances.

Remetendo a tabela 2x2, vamos novamente simular uma situação de exposição a um fator de risco e manifestação da doença. Observe abaixo como se apresentam as interseções:

- Pessoas que manifestaram a doença e foram expostas ao fator de risco: letra "`a`";
- Pessoas que **não** manifestaram a doença e foram expostas ao fator de risco: letra "`b`";
- Pessoas que manifestaram a doença e **não** foram expostas ao fator de risco: letra "`c`";
- Pessoas que **não** manifestaram a doença e **não** foram expostas ao fator de risco: letra "`d`".

Logo, a chance de ter sido exposto dado que manifestou a doença é representada por `a / c` e a chance de ter sido exposto dado que não manifestou a doença é representada por `b / d`.

Figura 5: Tabela 2x2 para avaliação do *Odds Ratio*.

Grupo	Doente	Não doente
Exposto	a	b
Não exposto	c	d
Odds	a / c	b / d

A partir da tabela acima, o cálculo da OR, envolvendo as odds, pode ser expresso matematicamente da seguinte forma:

$$Odds\ Ratio = \frac{\frac{a}{c}}{\frac{b}{d}} = \frac{a \times d}{b \times c}$$

Quando **o resultado dos cálculos da OR for igual a 1**, indicará a ausência de associação entre exposição e doença,

Quando a exposição ao fator de investigação aumenta a chance de se apresentar a doença, a OR **será maior que 1**, e

Quando a exposição ao fator age negativamente na doença, comportando-se como um possível fator de proteção, a OR **será menor que 1**.

Vamos praticar! Considere que enquanto profissional de vigilância, **você precisará avaliar a situação da coinfecção entre tuberculose e HIV em casos do Estado do Acre!** O adoecimento de tuberculose em pessoas vivendo com HIV (PVHIV) pode ser maior do que quando comparamos com outras pessoas não infectadas pelo vírus do HIV, principalmente pela influência da imunossupressão tornando esse grupo um dos mais vulneráveis, dentre os casos de tuberculose.

Levando em consideração esta premissa, você deverá analisar se há associação entre as pessoas com diagnóstico de tuberculose que vivem com HIV e o classificação do encerramento dos casos (desfecho). Para isso, imagine que você exportou o banco de dados de tuberculose do Sinan Net contendo todos os casos notificados entre anos de 2006 e 2020 no Acre. Este banco foi exportada no formato de arquivo "`.dbf`".

Nesta situação, as pessoas já apresentaram a doença. Afinal, a notificação de casos de tuberculose se dá a partir da confirmação e não da suspeita. Dessa forma, precisamos identificar uma associação entre pessoas que manifestaram a tuberculose (doença) e estiveram ou não expostos ao fator de risco (HIV). Conduziremos os cálculos para com o apoio da medida de associação odds *ratio*.

Acompanhe o passo a passo da construção desta análise no *script* abaixo e replique-o no seu `RStudio`:

1. Primeiro, iremos importar os dados {`base_tb_ac.dbf`} que serão analisados para o ambiente do `RStudio`. Estes estão no menu lateral "Arquivos", do módulo. Observe o script abaixo e replique o código em seu `RStudio`:

```
# Importando os dados {`base_tb_ac.dbf`} e armazenando
# no objeto (dataframe) de nome {`base_tb`}
base_tb <- read.dbf(file = 'Dados/base_tb_ac.dbf')
```

2. Para facilitar nossa análise, selecione apenas as variáveis relacionadas ao objetivo. Utilize, portanto, a função `select()` do pacote `dplyr` para selecionar estas variáveis. Consultando o dicionário de dados do Sinan Net, serão necessárias para esta análise as variáveis:

- `SITUA_ENCE`, que corresponde à situação do caso no momento do encerramento do caso e;
- `HIV`, que se refere ao resultado da sorologia para o vírus da imunodeficiência adquirida.

Acompanhe o código abaixo, e replique-o em seu `RStudio`:

```
# Selecionando apenas as colunas HIV e SITUA_ENCE
base_tb <- base_tb |> select(HIV, SITUA_ENCE)
```

3. Agora, para comparar os grupos categorize as variáveis selecionadas. No dicionário de dados da tuberculose, percebe-se que a variável `SITUA_ENCE` possui 10 categorias codificadas conforme a tabela abaixo. Dessa forma, categorize em 2 grupos considerando o seguinte:

- Registros de casos codificados como "1" serão categorizados como **Desfecho favorável** e;
- Registros de casos codificados como "2", "3", "7", "9" e "10" serão categorizados como **Desfecho desfavorável**.
- Registros de casos com as demais evoluções não serão consideradas por não estarem relacionadas diretamente à tuberculose.

Observe na Figura 6 como ficou a categorização da variável `SITUA_ENCE` em formato de tabela, a partir das definições pactuadas acima. Observe a transformação do número dos códigos para as suas respectivas categorias:

Figura 6: Tabela de categorização da variável `SITUA_ENCE`.

Código	Descrição	Categorização
1	Cura	Desfecho favorável
2	Abandono	Desfecho desfavorável
3	Óbito por TB	Desfecho desfavorável
4	Óbitos por outras causas	-
5	Transferência	-
6	Mudança de diagnóstico	-
7	TB-DR (multiresistente)	Desfecho desfavorável
8	Mudança de esquema	-
9	Falência	Desfecho desfavorável
10	Abandono primário	Desfecho desfavorável

4. Agora, vamos categorizar também a variável `HIV`, conforme tabela abaixo (Figura 7). Observe que iremos descartar as variáveis que não revelam o resultado da sorologia. Pos estamos fazendo uma avaliação dos desfechos que aconteceram logo os casos em brancos não serão utilizados.

Figura 7: Tabela de categorização da variável `HIV`.

Código	Descrição	Categorização
1	Positivo	Positivo
2	Negativo	Negativo
3	Em andamento	-
4	Não realizado	-

5. Utilizaremos a função `case_when()` dentro da função `mutate()` para ambas categorizações.
6. Perceba que, na sequência, também faremos a transformação das variáveis para fatores com seus respectivos níveis (*levels*).

Acompanhe com atenção o código abaixo referente ao passos de 1 a 6 e reproduza este script no seu `RStudio`:

```
# Atualizando o objeto base_tb com as alterações que serão feitas
base_tb <- base_tb |>

  # Utilizando a função mutate() para criar novas colunas
  mutate(

   # Transformando os códigos de situação de encerramento nos nomes correspondentes com a função case_when()
    SITUA_ENCE = case_when(
      SITUA_ENCE == "1" ~ "Favorável",
      SITUA_ENCE %in% c("2", "3", "7", "9", "10") ~ "Desfavorável",
      TRUE ~ NA_character_
    ),

   # Transformando os códigos de resultado de HIV nos nomes correspondentes com a função case_when()
    HIV = case_when(HIV == "1" ~ "Positivo",
                    HIV == "2" ~ "Negativo",
                    TRUE ~ NA_character_),

    # Convertendo as colunas SITUA_ENCE e HIV em variáveis do tipo factor
    SITUA_ENCE = factor(SITUA_ENCE, levels = c("Desfavorável",
"Favorável")),
    HIV = factor(HIV, levels = c("Positivo", "Negativo"))
  )
```

7. Agora, avalie a associação entre as variáveis por meio do teste estatístico qui-quadrado numa tabela 2x2:

- Definindo hipótese nula como não associação entre Sorologia de HIV e situação do encerramento e o nível de significância do teste estatístico em 0,05.

Acompanhe os *scripts* abaixo e replique-os em seu `RStudio`

```
base_tb |>

  # Criando uma tabela de resumo do cruzamento de informações sobre
  # resultado para HIV e situação de encerramento
  tbl_cross(
    row = HIV,
    col = SITUA_ENCE,
    percent = "row",
    missing = "no"
  ) |>

  # Adicionado o p-valor com a função add_p()
  add_p()
```

```
#> FALSE observations with missing data have been removed.
```

	SITUA_ENCE			
	Desfavorável	Favorável	Total	p-value1
HIV				<0.001
Positivo	16 (15%)	92 (85%)	108 (100%)	
Negativo	251 (6.0%)	3,907 (94%)	4,158 (100%)	
Total	267 (6.3%)	3,999 (94%)	4,266 (100%)	

[1] Pearson's Chi-squared test

Pronto! Terminamos a primeira etapa do estudo de *Odds Ratio*. Agora vamos lá responder à pergunta: **Existe associação entre pessoas com diagnóstico de tuberculose e vivem com HIV e a situação do caso no encerramento?**

Analisando a tabela, é possível notar que na última linha, a porcentagem de desfechos desfavoráveis é de 6,3% mas, quando a sorologia de HIV é positiva (ou seja, a pessoa foi exposta ao fator de risco), esse valor vai para 15%. Analisando por outro lado, o desfecho favorável para tuberculose, que é a Cura, diminui quando a sorologia da pessoa é positiva para HIV. Constatamos assim, que há diferença entre os grupos avaliados.

Nota-se ainda que o valor de `p` foi menor que 0,05, evidenciando que a hipótese nula pode ser rejeitada. Isso significa que há evidência para associação entre desfecho desfavorável para pessoas com tuberculose e quando expostas ao vírus HIV.

8. Seguindo a análise, precisaremos mensurar a magnitude dessa associação. Para o cálculo da odds ratio vamos utilizar a função `oddsratio()` do pacote `epitools`. Essa função é muito parecida com a função apresentada anteriormente e possui os mesmos argumentos principais:

    - `x`: vetor contendo os números da tabela 2x2;
    - `method`: método para calcular a *odds ratio*. Usaremos o método de Wald;
    - `rev`: ordem das linhas e colunas que serão realizados os cálculos. Usaremos "both".

Vamos inserir o vetor com os dados da tabela 2x2 acima, sem os totais. Vamos salvar o resultado em um objeto chamado `resultado_or_tb`.

Replique o *script* abaixo em seu `RStudio`:

```
# Calculando o oddsratio com a função oddsratio()
resultado_or_tb <- oddsratio(x = c(16, 92, 251, 3907),
                             method = "wald",
                             rev = "both")
```

9. Agora vamos visualizar os resultados do cálculo da odds ratio:

Acompanhe o script abaixo e replique em seu `RStudio`:

```
# Selecionando o objeto "measure" da lista resultado_or_tb
resultado_or_tb[["measure"]]
```

```
#>           odds ratio with 95% C.I.
#> Predictor  estimate    lower    upper
#>   Exposed2 1.000000       NA       NA
#>   Exposed1 2.707085 1.568086 4.673409
```

Considerando os padrões da tabela 2x2, a primeira linha foi a referência considerada para o cálculo dos expostos. Assim, **os cálculos nos levam a concluir que parece haver uma forte associação entre pessoas com tuberculose vivendo com HIV e desfecho desfavorável de tuberculose**, pois o valor da OR é maior que 1. Ou seja, pessoas que vivem com HIV apresentaram 2,7 vezes a chance de terem desfechos desfavoráveis de tuberculose em comparação àqueles que não foram expostos ao vírus.

Próximos cursos

Pronto, chegamos ao final do último módulo deste curso! Agora você já conhece as principais ações para automatizar suas análises de dados do dia a dia com o apoio da linguagem de programação `R`. Quer seguir a diante no aprendizado? Você encontrará outras etapas para aprofundamento das análises de dados em vigilância em saúde nos outros cursos. Aproveite e já faça sua inscrição nos cursos abaixo clicando nos links:

- Visualização de dados de interesse para a vigilância em saúde.
- Produção automatizada de relatórios na vigilância em saúde.
- Construção de diagramas de controle na vigilância em saúde.
- Linkage de bases de dados de saúde.
- Análise espacial de dados para a vigilância em saúde.
- Construção de paineis (dashboards) para monitoramento de indicadores de saúde.

Instituto Todos pela Saúde

ABRASCO
ASSOCIAÇÃO BRASILEIRA DE SAÚDE COLETIVA

UNA-SUS
Universidade Aberta do SUS

FIOCRUZ

UFPEL

UFSC