

Curso Intermediário (apoiado por *software* R) da Análise da Situação de Saúde aplicado a Emergências Sanitárias, com foco na COVID-19

Aula 03- Importando e padronizando dados no R

Apresentação

Nessa aula apresentaremos a primeira atividade prática do R. Estamos disponibilizando um recorte dos dados dos principais sistemas de notificação para Covid-19.

Para os casos de Síndrome Gripal (SG), estamos disponibilizando um recorte de casos do e-SUS Notifica para casos registrados no Distrito Federal. Para os casos de Síndrome Respiratória Aguda Grave (SRAG), estamos utilizando dados do SIVEP-Gripe. O recorte de dados é pequeno, para permitir que computadores com uma memória menor consigam realizar os exercícios de forma completa.

Nessa aula, vamos apresentar conceitos de padronização de dados, com dados das duas bases de dados. Vamos conhecer os tipos de variáveis do R e como transformar as variáveis para que a análise de dados seja realizada de forma adequada. Para isso, usaremos o “megapacote” de *Data Science* do R, o *Tidyverse*.

1. Introdução

Para a análise de dados em saúde, é muito importante realizar o tratamento de dados. Isso significa avaliar sua qualidade, entender que valores são discrepantes ou impossíveis e quais as maneiras em que se pode tratá-los. Após a importação dos dados para o *software* de análise, é importante analisar as variáveis, suas características, assim como o tipo de dado de cada uma delas.

O R traz funções que permitem a análise de dados desde a sua importação, até a curadoria e tratamento dos dados. O que fazer quando faltam dados? O que fazer quando os dados estão incorretos? Embora as boas práticas de vigilância nos instrua a coletar os dados da forma mais precisa possível, quando trabalhamos com dados de grandes bases de dados, cuja coleta não está sob nossa responsabilidade, é possível encontrar muitas inconsistências.

Para o tratamento de dados, introduziremos nessa aula o pacote Tidyverse (1). O Tidyverse é um pacote do R que inclui em si diversas bibliotecas de tratamento e análise de dados. O pacote é voltado para data science e inclui funções de tratamento, análise e gráficos. Para usar as diversas funcionalidades do Tidyverse, é necessário apenas instalar o pacote em seu R local.

O Tidyverse inclui os seguintes pacotes:

- Ggplot2 – biblioteca para a criação de gráficos;
- Dplyr – biblioteca para manipulação de dados;
- Tidyr - biblioteca para a organização de dados;
- Readr – biblioteca para a importação de dados;
- Purr – biblioteca para o tratamento de objetos em funções e vetores;
- Tibble – biblioteca para tratamento de objetos em formato de *data.frame*;
- Stringr – biblioteca para tratamento de objetos que apresentam variáveis em caracteres;
- Forcats – biblioteca para o tratamento de variáveis que se apresentam em formato de fatores (variáveis categóricas nominais ou ordinais);

2. Instalando e ativando o *Tidyverse*

Ao instalar o *Tidyverse* todas as funções dos pacotes citados são automaticamente incorporadas ao seu ambiente R. Apesar disso, você deverá

ativar o pacote todas as vezes que for utilizar uma função que pertence a cada uma de suas bibliotecas.

A instalação deverá ocorrer conforme os comandos:

```
install.packages("tidyverse")  
library(tidyverse)
```

3. Importando os dados

Disponibilizamos uma pequena amostra de dois bancos de dados públicos que se propõem a monitorar Covid-19, o e-SUS Notifica e o Sivep-Gripe. Os arquivos estão disponíveis juntamente com o material de apoio da aula. Ambos os arquivos estão em formato “.csv” e podem ser importados para o R por meio do comando de importação de arquivos desse formato.

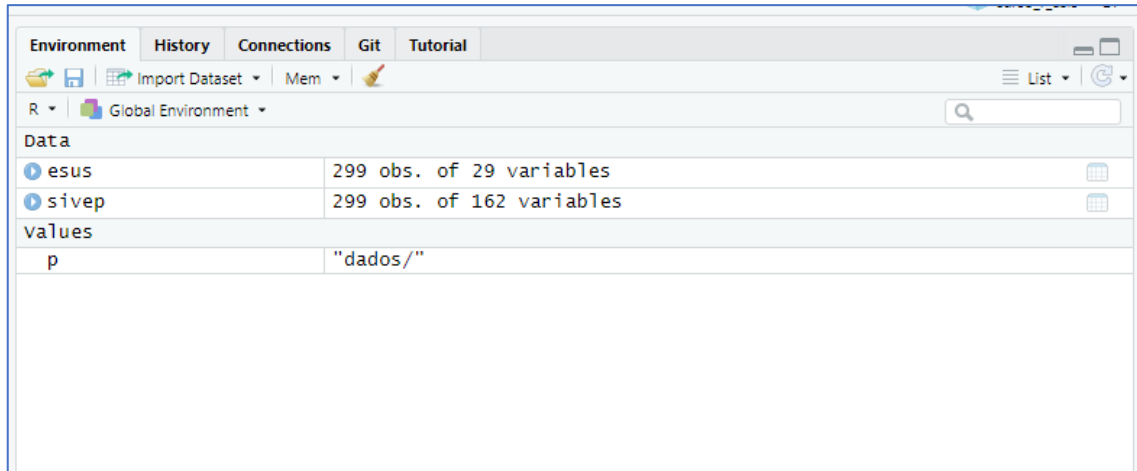
É importante destacar que o R possui pacotes e comandos para a importação de dados em quaisquer formatos. É possível importar dados de arquivos de planilhas (ods ou xlsx), em formato json, em formato dbf, em formato dbc, entre outros formatos. Caso você tenha necessidade de importar dados em um formato diferente, você pode consultar a Documentação do R (2) ou a comunidade de programação StackOverflow (3).

Para iniciar nossas atividades práticas, você deverá baixar a pasta do projeto R que usaremos nesse curso. Lembre-se, a pasta contém todos os arquivos e scripts que serão utilizados em todas as aulas. Usando o projeto R, você não terá dificuldades para acessar a pasta com os dados que usaremos nessa aula. Ao baixar e descompactar a pasta, clique duas vezes no arquivo .Rproject, conforme a imagem.

Para a importação nessa aula, usaremos dois arquivos: “20210601_dadosesus_df.csv” e “20210823_dadossivep.csv”, que foram disponibilizados em nosso ambiente de aprendizagem e possuem 300 observações cada. Para importar dados em .csv, é necessário usar os seguintes comandos:

```
p = "dados/" #pasta em que os dados se encontram  
esus = read.csv2(paste0(p, "20210601_dadosesus_df.csv"))  
sivep = read.csv2(paste0(p, "20210823_dadossivep.csv"))
```

É importante compreender se os dados foram importados para o ambiente R. Para isso, vamos apresentar duas diferentes técnicas. Utilizando o R Studio, observe se os objetos “esus” e “sivep” estão carregados na janela “Environment”, conforme a imagem.



Pelas linhas de comando você poderá utilizar o comando *glimpse*. Esse comando é aplicado a *data.frames*. Os objetos *data.frame* possuem linhas e colunas, em que as colunas são as variáveis ou atributos e as linhas são as observações.

```
glimpse(esus)
glimpse(sivep)
```

Esse comando permite conhecer o número de linhas e colunas de um *data.frame*. Além disso, permite visualizar cada uma das variáveis, mostrando uma amostra de suas observações e sua classe. Descrevemos as principais

- Data (date): variáveis que estão armazenadas em formato temporal;
- Caractere (character): variáveis que estão armazenadas em formato de palavras, ou seja, variáveis qualitativas nominais;
- Fator (factor): variáveis que estão armazenadas em formato de palavras, ou seja, variáveis qualitativas. Nesse caso é possível atribuir níveis para essas variáveis, que podem estar armazenadas como variáveis qualitativas ordinais;
- Numérico (numeric): variáveis que estão armazenadas em formato de número, seja discreto ou contínuo;

- Inteiro (integer): variáveis armazenadas como valores discretos, ou seja, valores inteiros;
- Contínuo (double): variáveis que são armazenadas como valores contínuos;

Ainda há outras classes de variáveis no R, que podem ser incorporadas por outros pacotes, entretanto compreender essas classes é suficiente para analisar os dados de bancos de dados em saúde.

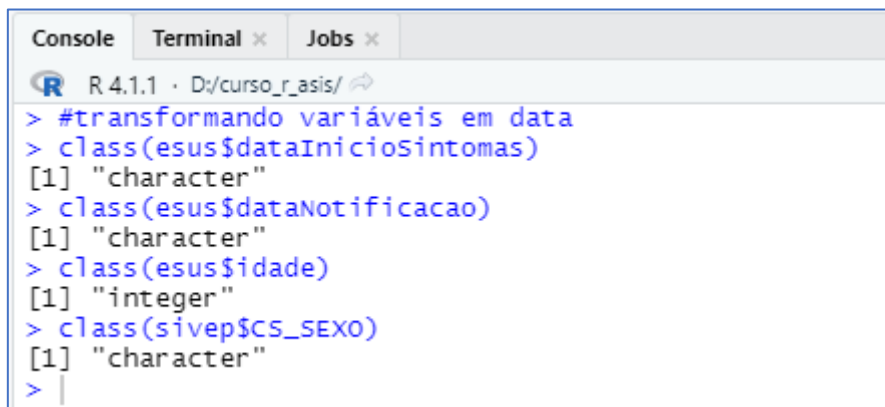
4. Tratamento de dados

Quando importamos dados para o R, o *programa* atribui automaticamente classes para cada uma das variáveis que foram importadas para o objeto *data.frame* no programa. Isso pode prejudicar análises de dados que você queira realizar, pois o R não conseguirá processar cálculos numéricos com variáveis em formato de palavra, por exemplo.

Para identificar uma variável você poderá usar, além do comando *glimpse*, apresentado anteriormente, o comando *class*, que irá permitir que você identifique o tipo para um objeto ou variável específica.

```
class(esus$dataInicioSintomas)
class(esus$dataNotificacao)
class(esus$idade)
class(sivep$CS_SEXO)
```

Nesse espaço, estamos identificando a classe de quatro variáveis. Ao utilizar esses comandos, recebemos o seguinte *output* do R:



```
Console Terminal x Jobs x
R 4.1.1 · D:/curso_r_asis/
> #transformando variáveis em data
> class(esus$dataInicioSintomas)
[1] "character"
> class(esus$dataNotificacao)
[1] "character"
> class(esus$idade)
[1] "integer"
> class(sivep$CS_SEXO)
[1] "character"
> |
```

Vamos tratar esses dados para transformar as variáveis em formato de data, fator e para investigar quais valores são muito discrepantes e que podem estar incorretos, enviando possíveis análises.

a. Tratando datas

Nós investigamos a classe de datas da base de dados do e-SUS Notifica e do Sivep-Gripe. Para todas essas variáveis recebemos o *output* (mostrado acima) de que os dados estão em formato de “*character*”, ou seja, o R entende que esses dados são palavras.

Para transformar esses dados em datas e conferir se foram transformados, vamos usar os seguintes comandos:

```
esus$dataInicioSintomas = as.Date(esus$dataInicioSintomas)
esus$dataNotificacao = as.Date(esus$dataNotificacao)

class(esus$dataNotificacao)
class(esus$dataInicioSintomas)

sivep$DT_NOTIFIC = as.Date(sivep$DT_NOTIFIC, "%d/%m/%Y")
sivep$DT_SIN_PRI = as.Date(sivep$DT_SIN_PRI, "%d/%m/%Y")

class(sivep$DT_NOTIFIC)
class(sivep$DT_SIN_PRI)
```

Após isso, vamos receber os seguintes *outputs*:

```
> class(esus$dataInicioSintomas)
[1] "Date"
> class(esus$dataNotificacao)
[1] "Date"
> class(esus$idade)
[1] "integer"
> class(sivep$CS_SEXO)
[1] "character"
>
> esus$dataInicioSintomas = as.Date(esus$dataInicioSintomas)
> esus$dataNotificacao = as.Date(esus$dataNotificacao)
>
> class(esus$dataNotificacao)
[1] "Date"
> class(esus$dataInicioSintomas)
[1] "Date"
>
> sivep$DT_NOTIFIC = as.Date(sivep$DT_NOTIFIC, "%d/%m/%Y")
> sivep$DT_SIN_PRI = as.Date(sivep$DT_SIN_PRI, "%d/%m/%Y")
>
> class(sivep$DT_NOTIFIC)
[1] "Date"
> class(sivep$DT_SIN_PRI)
[1] "Date"
> |
```

Ao testar novamente a classe das variáveis percebemos que elas foram transformadas em variáveis do tipo data. Após essa transformação, vamos verificar as datas das duas bases para verificar se existem valores discrepantes ou incorretos. Para isso, vamos considerar as datas de Notificação maiores que o dia 26 de fevereiro de 2020, quando o primeiro caso de Covid-19 foi confirmado

no país e as datas de disponibilização dos dados, sendo o dia 01 de junho de 2021 para o e-SUS Notifica e o dia 23 de agosto de 2021 para o Sivep-Gripe. Caso os valores da data de notificação forem menores que a data do primeiro caso ou maior que a data de disponibilização dos dados, consideraremos que os dados estão incorretos.

Criaremos um histograma com os dados e usaremos a opção *summary* para visualizar as medidas de posição desses dados e compreender se há muitos valores discrepantes.

```
summary(esus$dataInicioSintomas)
hist(esus$dataInicioSintomas, breaks = 30)

esús$dataInicioSintomas = as.Date(ifelse(esus$dataInicioSintomas <= "2020-03-01" | esus$dataInicioSintomas > "2021-06-01",
                                          NA, esus$dataInicioSintomas), origin = "1970-01-01")

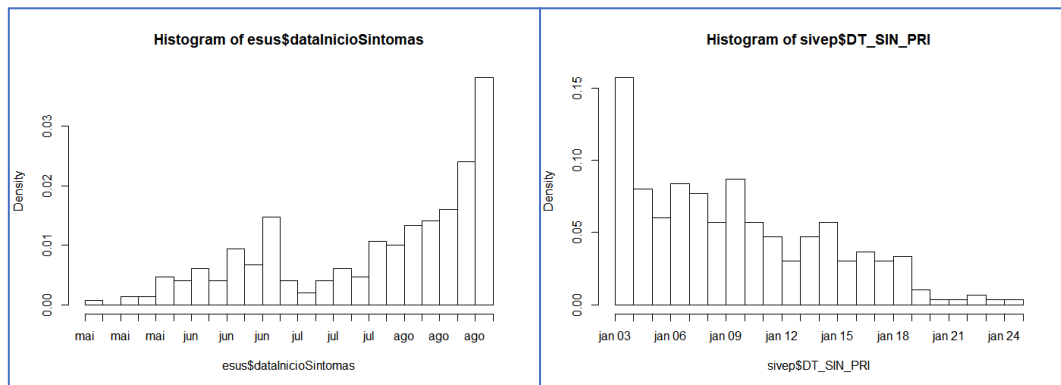
summary(esus$dataInicioSintomas)
hist(esus$dataInicioSintomas, breaks = 30)

summary(sivep$DT_SIN_PRI)
hist(sivep$DT_SIN_PRI, breaks = 30)
```

Ao analisar os dados, temos os seguintes *outputs*:

```
> summary(esus$dataInicioSintomas)
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
"2020-05-10" "2020-06-29" "2020-08-09" "2020-07-29" "2020-08-24" "2020-08-31"
> hist(esus$dataInicioSintomas, breaks = 30)
>
> esús$dataInicioSintomas = as.Date(ifelse(esus$dataInicioSintomas <= "2020-03-01" | esus$dataInicioSintomas > "2021-06-01",
+                                          NA, esus$dataInicioSintomas), origin = "1970-01-01")
> summary(esus$dataInicioSintomas)
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
"2020-05-10" "2020-06-29" "2020-08-09" "2020-07-29" "2020-08-24" "2020-08-31"
> hist(esus$dataInicioSintomas, breaks = 30)
>
> summary(sivep$DT_SIN_PRI)
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
"2021-01-03" "2021-01-06" "2021-01-09" "2021-01-09" "2021-01-14" "2021-01-25"
> hist(sivep$DT_SIN_PRI, breaks = 30)
> |
```


Para os histogramas possuímos os seguintes valores:



Não encontramos valores discrepantes na base de dados. De qualquer forma usamos a função do terceiro comando para excluir os dados que mostravam valores discrepantes segundo a regra que criamos.

b. Tratando variáveis numéricas

Para variáveis numéricas os tratamentos, em geral, são mais simples que o tratamento para outros tipos de variáveis. Nesse caso usaremos comandos simples para transformar os dados de idade em variáveis numéricas e depois remover os valores muito discrepantes, que aqui consideraremos os valores maiores que 100.

Os comandos são os seguintes:

```
esus$idade = as.numeric(esus$idade)
summary(esus$idade)
hist(esus$idade)

esus$idade = ifelse(esus$idade > 100, NA, esus$idade)
summary(esus$idade)
hist(esus$idade)
```

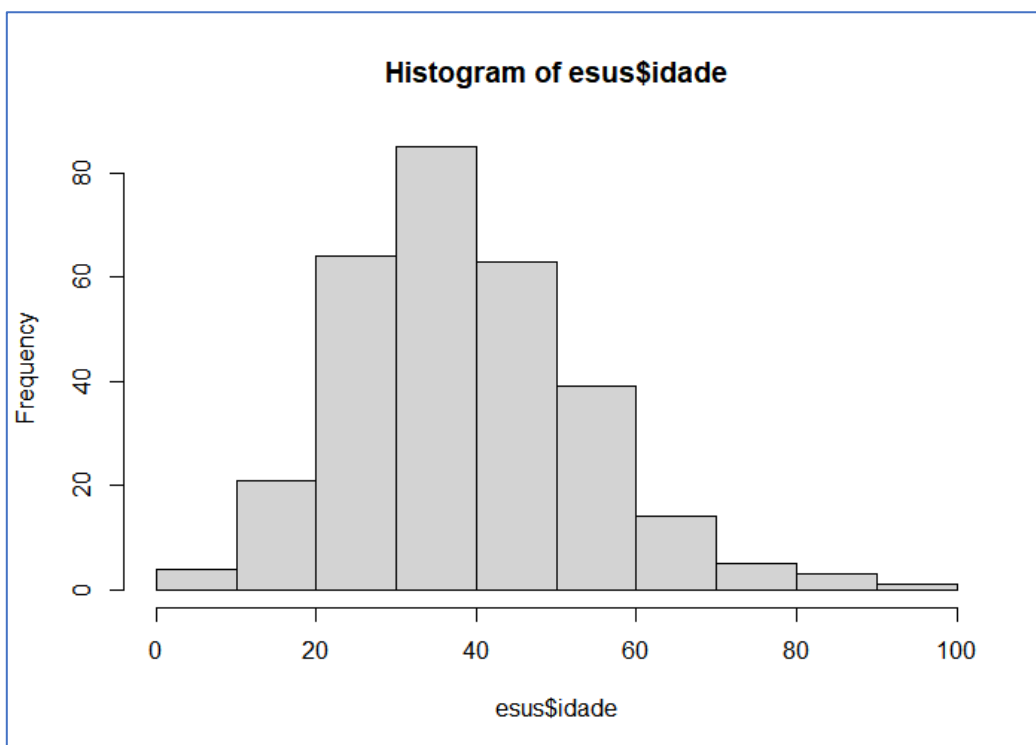
Nesse caso, estamos transformando as variáveis em número e usando o comando *summary* e *hist* para olhar suas medidas de posição e visualizar sua distribuição.

Com os resultados, conseguimos identificar que não há valores superiores a 100 no campo de idade. De qualquer forma, incluímos os comandos. Nós obtemos os seguintes outputs:

```

> #ajustando idade como número
> esus$idade = as.numeric(esus$idade)
> summary(esus$idade)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.00  28.50   37.00   38.94   47.00   91.00
> hist(esus$idade)
> esus$idade = ifelse(esus$idade > 100, NA, esus$idade)
> summary(esus$idade)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.00  28.50   37.00   38.94   47.00   91.00
> hist(esus$idade)
> |

```



c. Tratando variáveis categóricas

Em geral não é necessário realizar o tratamento de variáveis categóricas, pois a maioria das análises epidemiológicas são numéricas. Apesar disso, há diversas funções para transformar variáveis categóricas.

Nesse exercício, criaremos *levels* para que a variável sexo apareça na ordem M, F. Para isso, usaremos os seguintes comandos:

```

table(sivep$CS_SEX0)
sivep$CS_SEX0 = factor(sivep$CS_SEX0, levels = c("M", "F", "I"))
class(sivep$CS_SEX0)

```

Nesse caso, estamos transformando a variável em um fator, em que a variável aparecerá na ordem delimitada pela função, com os seguintes outputs:

```
> #tratando com variáveis nominais
> table(sivep$CS_SEXO)

  F    M
147 152
> sivep$CS_SEXO = factor(sivep$CS_SEXO, levels = c("M", "F", "I"))
> class(sivep$CS_SEXO)
[1] "factor"
> |
```

Assim, conseguimos criar tabelas padronizadas. É importante ressaltar que usamos essa variável apenas como exemplo. Em geral, usamos o argumento *levels* para variáveis categóricas ordinais, como faixa etária, escolaridade, faixa de renda entre outras.

Assim, encerramos essa aula! Na próxima aula abordaremos estatísticas descritivas! Até lá.

5. Referências Bibliográficas

1. Tidyverse. R packages for data science. Disponível em:
<<https://www.tidyverse.org/>> Acesso em: 30 ago 2021.
2. R Documentation. Disponível em: <<https://www.rdocumentation.org/>>.
Acesso em: 30 ago 2021.
3. StackOverflow em Português. Disponível em:
<<https://pt.stackoverflow.com/>>. Acesso em: 30 ago 2021.