



Universidade de São Paulo
Faculdade de Saúde Pública
Departamento de Epidemiologia

EPI5713- Introdução ao R para a Análise de Dados

Banco de Dados





- **Referências:**

- Wickham H, Grolemund G. R for Data Science. O'Reilly; 2017.
- Baquero OS. Manipulação e visualização de dados no R. disponível em: <https://github.com/leb-fmvz-usp/manipulacao-e-visualizacao-de-dados-no-r>
- Mello MP, Peternelli LA. Conhecendo o R: uma visão mais que estatística. Viçosa, MG: Ed. UFV, 2013.

“Visualization is an important tool for insight generation, but it is rare that you get the data in exactly the right form you need”

Wickham & Grolemund, 2017

Conteúdo

- O que é um data.frame?
- Criar um data.frame
- Alterar nomes de linhas ou colunas
- Acessar elementos de data.frames
- Transformar variáveis
- Adicionar ou eliminar linhas ou colunas
- Selecionar um subgrupo de um data.frame
- Ordenação de linhas
- Separar data.frame, segundo variáveis
- Agrupar data.frames
- Valores missing

Conteúdo

Utilizando o pacote dplyr:

- Selecionar observações por valor → `filter ()`
- Reordenar linhas → `arrange ()`
- Selecionar variáveis por nome → `select ()`
- Criar novas variáveis em função de outras → `mutate ()`
→ `group_by()`
- Relacionar data.frames (Mutating joins)

Data.frame

- Banco de dados
- 2 dimensões
- Difere de matrizes, pois as colunas podem armazenar diferentes tipos de elementos (numéricos ou caracteres)
- Forma de armazenar dados:
 - Cada linha corresponde a uma unidade (observação)
 - Cada coluna corresponde a uma variável observada em cada unidade
 - As colunas devem ter o mesmo tamanho

Criando data.frame

- Por meio de vetores:

obs	nome	idade	sexo	conceito	extra
1	ana	20	1	B	1
2	maria	25	1	C	0
3	carlos	35	2	A	1
4	ricardo	30	2	A	0
5	laura	43	1	D	0
6	rafael	32	2	B	1
7	cristina	31	1	B	0
8	priscila	23	1	A	1
9	luciana	29	1	C	0
10	jose	50	2	D	1

```
obs <- c(1,2,3,4,5,6,7,8,9,10)
```

```
nome <- c("ana","maria","carlos","ricardo".....)
```

Criando data.frame

- Utilizando o comando scan:

```
sexo <- scan ()
```

1
1
2
2
1
2
1
1
1
2

Copiar e colar da planilha

Enter

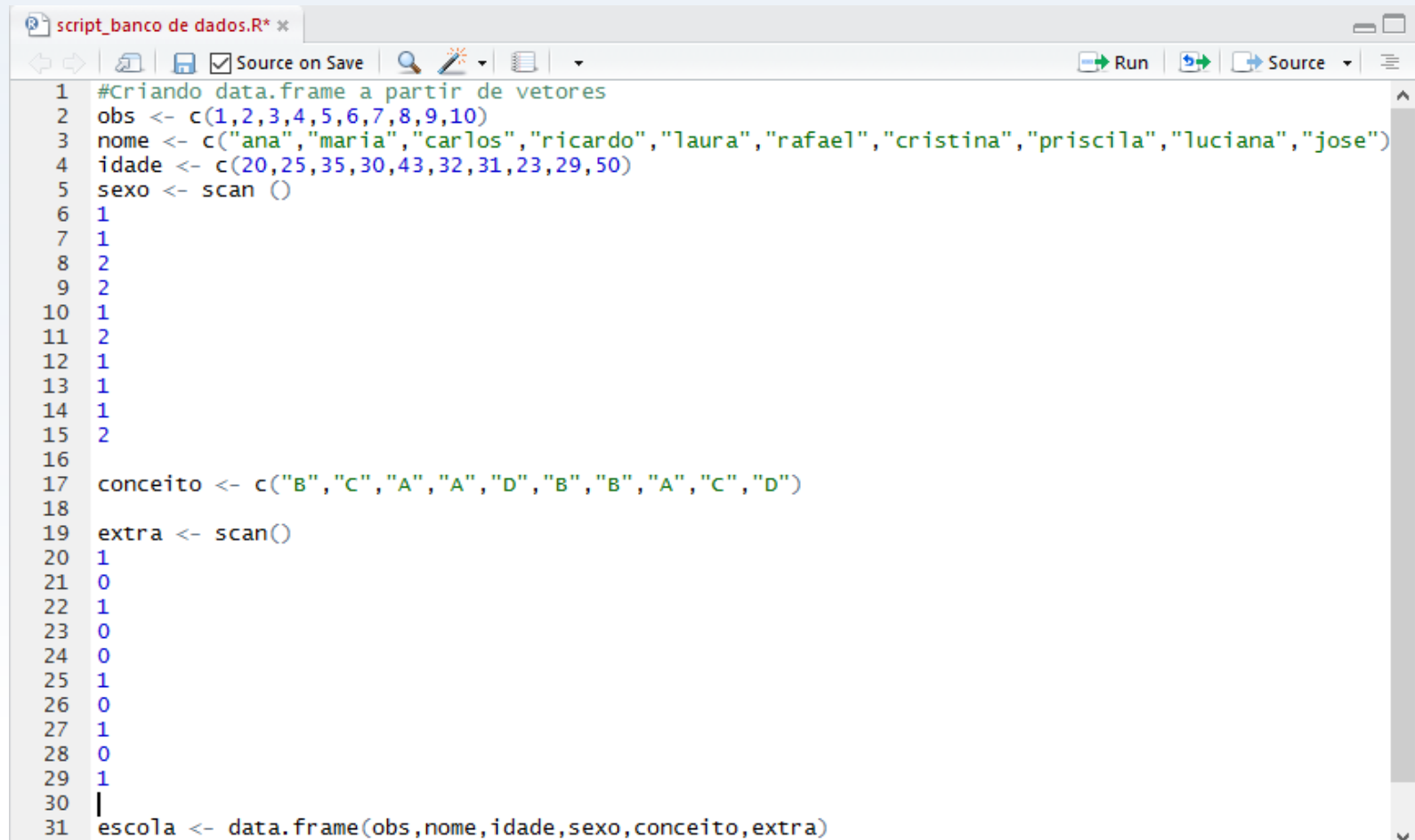
Criando data.frame

- Montar banco de dados a partir dos vetores criados:

`data.frame()`

```
escola <- data.frame(obs,nome,idade,sexo,conceito,extra)
```

Criando data.frame



The screenshot shows an R script editor window titled "script_banco de dados.R*". The script contains the following R code:

```
1 #Criando data.frame a partir de vetores
2 obs <- c(1,2,3,4,5,6,7,8,9,10)
3 nome <- c("ana","maria","carlos","ricardo","laura","rafael","cristina","priscila","luciana","jose")
4 idade <- c(20,25,35,30,43,32,31,23,29,50)
5 sexo <- scan ()
6 1
7 1
8 2
9 2
10 1
11 2
12 1
13 1
14 1
15 2
16
17 conceito <- c("B","C","A","A","D","B","B","A","C","D")
18
19 extra <- scan()
20 1
21 0
22 1
23 0
24 0
25 1
26 0
27 1
28 0
29 1
30 |
31 escola <- data.frame(obs,nome,idade,sexo,conceito,extra)
```

The script defines several vectors: `obs` (1 to 10), `nome` (names), `idade` (ages), `sexo` (sex, entered via `scan()`), `conceito` (grades), and `extra` (extra data, entered via `scan()`). Finally, it creates a `data.frame` named `escola` containing all these vectors.

Criando data.frame

Data	
escola	10 obs. of 6 variables
values	
conceito	chr [1:10] "B" "C" "A" "A" "D" "B" "B" "A" "C" "D"
extra	num [1:10] 1 0 1 0 0 1 0 1 0 1
idade	num [1:10] 20 25 35 30 43 32 31 23 29 50
nome	chr [1:10] "ana" "maria" "carlos" "ricardo" "laura" "rafael" "cristina" ...
obs	num [1:10] 1 2 3 4 5 6 7 8 9 10
sexo	num [1:10] 1 1 2 2 1 2 1 1 1 2

Verificar a estrutura

escola

```
> escola
```

	obs	nome	idade	sexo	conceito	extra
1	1	ana	20	1	B	1
2	2	maria	25	1	C	0
3	3	carlos	35	2	A	1
4	4	ricardo	30	2	A	0
5	5	laura	43	1	D	0
6	6	rafael	32	2	B	1
7	7	cristina	31	1	B	0
8	8	priscila	23	1	A	1
9	9	luciana	29	1	C	0
10	10	jose	50	2	D	1

View (escola)

str(escola)

```
> str(escola)
'data.frame': 10 obs. of 6 variables:
 $ obs      : num  1 2 3 4 5 6 7 8 9 10
 $ nome     : Factor w/ 10 levels "ana","carlos",...: 1 7 2 10 5 9 3 8 6 4
 $ idade    : num  20 25 35 30 43 32 31 23 29 50
 $ sexo     : num  1 1 2 2 1 2 1 1 1 2
 $ conceito : Factor w/ 4 levels "A","B","C","D": 2 3 1 1 4 2 2 1 3 4
 $ extra    : num  1 0 1 0 0 1 0 1 0 1
> |
```

Importação de banco de dados

```
escolaimp <- read.table ("banco de dados.csv", header=T,  
sep=";")
```

- escolaimp
- str(escolaimp)
- reader ()
- Pacotes específicos para ler arquivos:

Excel → xlsx read.xlsx()

Importação de banco de dados

- `read.dta`
- `install.packages("foreign")`
- `library (foreign)`
- `escoladta <- read.dta`
`("banco de dados.dta")`
- `View(escoladta)`

Alterar nomes de linhas/colunas

- Linhas → `row.names ()`
- Colunas → `colnames ()` ou `names ()`

Ex: `row.names (escola) <- c ("linha 1", "linha 2", "linha 3", "linha 4", "linha 5", "linha 6", "linha 7", "linha 8", "linha 9", "linha 10")`

`colnames (escola) <- c ("obs1", "nome1", "idade1", "sexo1", "conceito1", "extra1")`

`names (escola) <- c ("obs", "nome", "idade", "sexo", "conceito", "extra")`

Acessando elementos de um data.frame

- Há 2 formas:

1) Índice linha e coluna

escola [2,1] → elemento da linha 2 e coluna 1

escola [2,]

```
> escola [2,1]
[1] 2
> escola [2,]
      obs nome idade sexo conceito extra
linha 2   2 maria   25    1         C     0
```

2) Utilizando o símbolo \$

escola\$nome

escola\$nome [2]

escola\$nome [1:3]

Transformando variáveis

- `as.factor ()`
- `as.character ()`
- `as.numeric`

```
escola [,1] <- as.factor (escola[,1])
```

```
escola [,2] <- as.character (escola[,2])
```

```
escola [,1] <- as.numeric (escola[,1])
```

```
data.frame( ..... stringsAsFactors = F)
```

Conteúdo

- ✓ O que é um data.frame?
- ✓ Criar um data.frame
- ✓ Alterar nomes de linhas ou colunas
- ✓ Acessar elementos de data.frames
- ✓ Transformar variáveis
 - Adicionar ou eliminar linhas ou colunas
 - Selecionar um subgrupo de um data.frame
 - Ordenação de linhas
 - Separar data.frame, segundo variáveis
 - Agrupar data.frames
 - Valores missing

Adicionar/eliminar linhas/colunas

- Colunas → `cbind ()`
- Linhas → `rbind ()`

```
escola <- cbind (escola, nota = c(8,6,9,10,4,8,8,9,5,3))
```

```
escola <-rbind (escola, "linha 11" = c(11,"caio",33,2,"A",1,9))
```

Removendo

Colunas: `escola <- escola [, -7]`

Linhas: `escola <- escola[1-10,]`

Opção colunas: `escola$nota <- c(8,6,9,10,4,8,8,9,5,3))`

Selecionar 1 subgrupo

- `escola2 <- escola[1:6,]` (seleciona linhas 1 a 6)

```
> escola2 <- escola[1:6,]
> escola2
```

	obs	nome	idade	sexo	conceito	extra	nota
linha 1	1	ana	20	1	B	1	8
linha 2	2	maria	25	1	C	0	6
linha 3	3	carlos	35	2	A	1	9
linha 4	4	ricardo	30	2	A	0	10
linha 5	5	laura	43	1	D	0	4
linha 6	6	rafael	32	2	B	1	8

- `escola2 <- escola2 [, -5]` (exclui coluna 5)

```
> escola2 <- escola2 [, -5]
> escola2
```

	obs	nome	idade	sexo	extra	nota
linha 1	1	ana	20	1	1	8
linha 2	2	maria	25	1	0	6
linha 3	3	carlos	35	2	1	9
linha 4	4	ricardo	30	2	0	10
linha 5	5	laura	43	1	0	4
linha 6	6	rafael	32	2	1	8

- `escola2[escola2$sexo==1]` (exibe só feminino)

```
> escola2[escola2$sexo==1,]
      obs nome idade sexo extra nota
linha 1  1  ana   20    1     1     8
linha 2  2 maria  25    1     0     6
linha 5  5 laura  43    1     0     4
```

Ordenação de linhas

- `escola [order(escola$nota),]` (crescente por nota)
- `escola[rev(order(escola$nota),)]` (decrescente por nota)
- Como ordenar por obs (crescente)?

```
escola [order(escola$obs),]
```

Separando por grupos

- `split (escola,sexo)`

```
> split (escola,sexo)
```

```
$`1`
```

	obs	nome	idade	sexo	conceito	extra	nota
linha 1	1	ana	20	1	B	1	8
linha 2	2	maria	25	1	C	0	6
linha 5	5	laura	43	1	D	0	4
linha 7	7	cristina	31	1	B	0	8
linha 8	8	priscila	23	1	A	1	9
linha 9	9	luciana	29	1	C	0	5

```
$`2`
```

	obs	nome	idade	sexo	conceito	extra	nota
linha 3	3	carlos	35	2	A	1	9
linha 4	4	ricardo	30	2	A	0	10
linha 6	6	rafael	32	2	B	1	8
linha 10	10	jose	50	2	D	1	3

Agrupando data.frames

- merge ()
- Une dois banco de dados a partir de uma coluna em comum

Criar o 2º banco de dados:

```
estado <- c  
("SP","MT","RR","RJ","SP","ES","PE","AM","SC","GO")
```

```
novo <- data.frame(obs,nome,estado)
```

```
> novo  
  obs  nome estado  
1    1   ana     SP  
2    2  maria    MT  
3    3 carlos    RR  
4    4 ricardo   RJ  
5    5  laura    SP  
6    6 rafael    ES  
7    7 cristina  PE  
8    8 priscila  AM  
9    9  luciana  SC  
10   10   jose    GO
```

Agrupando data.frames

- `completo <- merge (escola,novo,by="obs")`

```
> completo
```

	obs	nome.x	idade	sexo	conceito	extra	nota	nome.y	estado
1	1	ana	20	1	B	1	8	ana	SP
2	2	maria	25	1	C	0	6	maria	MT
3	3	carlos	35	2	A	1	9	carlos	RR
4	4	ricardo	30	2	A	0	10	ricardo	RJ
5	5	laura	43	1	D	0	4	laura	SP
6	6	rafael	32	2	B	1	8	rafael	ES
7	7	cristina	31	1	B	0	8	cristina	PE
8	8	priscila	23	1	A	1	9	priscila	AM
9	9	luciana	29	1	C	0	5	luciana	SC
10	10	jose	50	2	D	1	3	jose	GO

- `completo <- merge (escola,novo,by="nome")`

Valores missing (NA)

- Inf, -Inf, NaN – not a number, NA – not available
- Exemplo: situação conjugal (conj)

```
conj <- c(1,2,NA,1,1,2,NA,1,1,2)
```

Se codificar meus valores missing como 99, 9999, 0 etc

```
completo3 <- cbind (completo, conj=c(1,2,99,1,1,2,99,1,1,2))
```

```
completo3$conj[completo3$conj==99] <- NA
```

```
completo3
```

Carregando banco do R

- `help(datasets)`
- `library(help = "datasets")`

- `data (Titanic)`
- `help(Titanic)`

- `str(Titanic)`

Conteúdo

- ✓ O que é um data.frame?
- ✓ Criar um data.frame
- ✓ Alterar nomes de linhas ou colunas
- ✓ Acessar elementos de data.frames
- ✓ Transformar variáveis
- ✓ Adicionar ou eliminar linhas ou colunas
- ✓ Selecionar um subgrupo de um data.frame
- ✓ Ordenação de linhas
- ✓ Separar data.frame, segundo variáveis
- ✓ Agrupar data.frames
- ✓ Valores missing

Conteúdo

Utilizando o pacote dplyr:

- Selecionar observações por valor → `filter ()`
- Reordenar linhas → `arrange ()`
- Selecionar variáveis por nome → `select ()`
- Criar novas variáveis em função de outras → `mutate ()`
→ `group_by()`
- Relacionar data.frames (Mutating joins)

Pacote dplyr

- Pré-requisitos:
→ Membro do tidyverse

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

```
library (tidyverse)
```

Pacote dplyr

- `filter ()` → seleciona obs pelos valores
- `arrange()` → reordena linhas
- `select ()` → seleciona variáveis
- `mutate ()` → cria novas variáveis em função de outras existentes
- `summarize ()` → múltiplos valores em um único

`group_by` → grupo por grupo

Pacote dplyr

1. O primeiro argumento é um data.frame
2. Os argumentos subsequentes descrevem o que fazer com o data.frame, usando nome de variáveis (sem " ")
3. O resultado é um novo data.frame

```
extra1 <- filter(escola, extra == 1)
```



```
> extra1
```

	obs	nome	idade	sexo	conceito	extra
1	1	ana	20	1	B	1
2	3	carlos	35	2	A	1
3	6	rafael	32	2	B	1
4	8	priscila	23	1	A	1
5	10	jose	50	2	D	1

Comparações / Operadores

> (maior)

>= (maior ou igual)

< (menor)

<= (menor ou igual)

!= (diferente)

== (igual)

& (AND)

| (OR)

! (NOT)

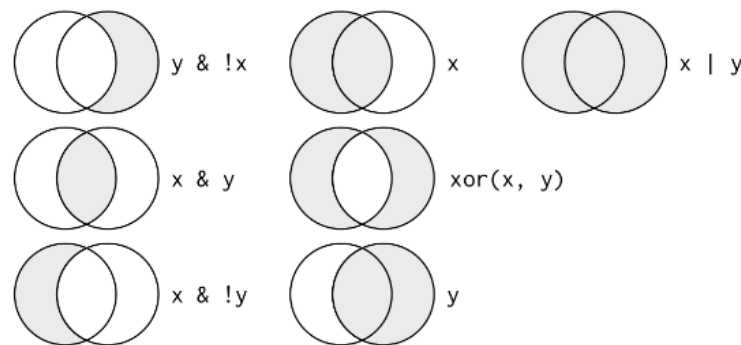


Figure 5.1: Complete set of boolean operations. x is the left-hand circle, y is the right-hand circle, and the shaded region show which parts each operator selects.

filter ()

filter(banco de dados, variável - critério)

Ex:

filter(vacina, escola==1,vacinado==1)

```
> filter(vacina, escola==1,vacinado==1)
# A tibble: 11 x 7
   id      dtnasc  dtentrev vacinado escola idademaescmae
  <dbl>  <dtm>    <dtm>    <dbl>  <dbl>  <dbl>  <dbl>
1     1 2015-01-01 2017-06-01     1      1     18     1
2     3 2015-02-03 2017-06-01     1      1     25     2
3     5 2015-01-05 2017-06-01     1      1     26     4
4     6 2015-01-16 2017-06-01     1      1     30     3
5     7 2015-05-07 2017-06-01     1      1     25     2
6    14 2015-03-14 2017-06-01     1      1     34     3
7    17 2015-05-07 2017-06-01     1      1     37     4
8    18 2015-05-28 2017-06-01     1      1     38     3
9    20 2015-10-12 2017-06-01     1      1     25     3
10   21 2015-01-01 2017-06-01     1      1     35     4
11   22 2015-04-16 2017-06-01     1      1     35     4
```

filter ()

Ex:

```
filter(vacina, escola==1 & vacinado==1)
```

```
> filter(vacina, escola==1 & vacinado==1)
# A tibble: 11 x 7
   id      dtnasc    dtentrev vacinado escola idademaescmae
   <dbl>   <dtm>      <dtm>    <dbl>   <dbl>   <dbl>  <dbl>
1     1 2015-01-01 2017-06-01     1     1     18     1
2     3 2015-02-03 2017-06-01     1     1     25     2
3     5 2015-01-05 2017-06-01     1     1     26     4
4     6 2015-01-16 2017-06-01     1     1     30     3
5     7 2015-05-07 2017-06-01     1     1     25     2
6    14 2015-03-14 2017-06-01     1     1     34     3
7    17 2015-05-07 2017-06-01     1     1     37     4
8    18 2015-05-28 2017-06-01     1     1     38     3
9    20 2015-10-12 2017-06-01     1     1     25     3
10   21 2015-01-01 2017-06-01     1     1     35     4
11   22 2015-04-16 2017-06-01     1     1     35     4
```

filter ()

Ex:

```
filter(vacina, escola==1 & idademaes<20)
```

```
> filter(vacina, escola==1 & idademaes<20)
# A tibble: 2 x 7
   id      dtnasc   dtentrev vacinado escola idademaes escmaes
  <dbl>   <dtm>     <dtm>    <dbl>   <dbl>   <dbl>   <dbl>
1     1 2015-01-01 2017-06-01     1     1      18     1
2     2 2015-02-02 2017-06-01     0     1      19     1
```

arrange ()

- arrange(vacina, idadema)

```
> arrange(vacina, idadema)
# A tibble: 100 × 7
   id      dtnasc   dtentrev vacinado escola idadema escmae
  <dbl>   <dtm>     <dtm>     <dbl>   <dbl>   <dbl>   <dbl>
1    55 2015-05-25 2017-06-01     1     3     14     1
2    54 2015-06-26 2017-06-01     1     3     15     1
3    56 2015-03-29 2017-06-01     1     3     15     1
4    57 2015-02-20 2017-06-01     1     3     16     1
5    58 2015-07-17 2017-06-01     1     3     17     2
6     1 2015-01-01 2017-06-01     1     1     18     1
7    53 2015-08-31 2017-06-01     0     3     18     1
8    59 2015-08-18 2017-06-01     1     3     18     2
9    60 2015-08-22 2017-06-01     1     3     18     2
10     2 2015-02-02 2017-06-01     0     1     19     1
# ... with 90 more rows
```

arrange ()

- `arrange(vacina, desc(idademaes))`

```
> arrange(vacina, desc(idademaes))
# A tibble: 100 x 7
   id      dtnasc    dtentrev vacinado escola idademaes escmaes
  <dbl>    <dtm>    <dtm>    <dbl>   <dbl>   <dbl>   <dbl>
1      8 2015-03-28 2017-06-01         0       1      45       1
2     30 2015-04-16 2017-06-01         0       2      43       3
3     29 2015-07-09 2017-06-01         1       2      41       4
4     72 2015-06-26 2017-06-01         0       3      40       4
5     48 2015-10-05 2017-06-01         1       2      39       3
6     81 2015-03-16 2017-06-01         1       3      39       2
7     18 2015-05-28 2017-06-01         1       1      38       3
8     47      <NA> 2017-06-01         1       2      38       3
9     71 2015-08-31 2017-06-01         1       3      38       4
10    80 2015-03-14 2017-06-01         1       3      38       2
# ... with 90 more rows
```

arrange ()

- arrange(vacina, id)

```
> arrange(vacina, id)
# A tibble: 100 × 7
   id      dtnasc    dtentrev vacinado escola idade_mae esc_mae
  <dbl>    <dtm>    <dtm>    <dbl>   <dbl>   <dbl>   <dbl>
1     1 2015-01-01 2017-06-01         1       1       18        1
2     2 2015-02-02 2017-06-01         0       1       19        1
3     3 2015-02-03 2017-06-01         1       1       25        2
4     4 2015-01-04 2017-06-01         0       1       35        3
5     5 2015-01-05 2017-06-01         1       1       26        4
6     6 2015-01-16 2017-06-01         1       1       30        3
7     7 2015-05-07 2017-06-01         1       1       25        2
8     8 2015-03-28 2017-06-01         0       1       45        1
9     9 2015-07-09 2017-06-01         0       1       27        4
10    10 2015-10-10 2017-06-01         0       1       28        3
# ... with 90 more rows
```


select ()

- select(vacina, 1:3)

```
> select(vacina, 1:3)
# A tibble: 100 × 3
      id      dtnasc      dtentrev
  <dbl>    <dtm>    <dtm>
1     1 2015-01-01 2017-06-01
2     2 2015-02-02 2017-06-01
3     3 2015-02-03 2017-06-01
4     4 2015-01-04 2017-06-01
5     5 2015-01-05 2017-06-01
6     6 2015-01-16 2017-06-01
7     7 2015-05-07 2017-06-01
8     8 2015-03-28 2017-06-01
9     9 2015-07-09 2017-06-01
10    10 2015-10-10 2017-06-01
# ... with 90 more rows
```

- select(vacina, c(id,dtnasc,dtentrev))

```
> select(vacina, c(id,dtnasc,dtentrev))
# A tibble: 100 × 3
      id      dtnasc      dtentrev
  <dbl>    <dtm>    <dtm>
1     1 2015-01-01 2017-06-01
2     2 2015-02-02 2017-06-01
3     3 2015-02-03 2017-06-01
4     4 2015-01-04 2017-06-01
5     5 2015-01-05 2017-06-01
6     6 2015-01-16 2017-06-01
7     7 2015-05-07 2017-06-01
8     8 2015-03-28 2017-06-01
9     9 2015-07-09 2017-06-01
10    10 2015-10-10 2017-06-01
# ... with 90 more rows
```

select ()

- select(vacina, id,dtnasc,dtentrev)

```
> select(vacina, id,dtnasc,dtentrev)
# A tibble: 100 × 3
      id      dtnasc    dtentrev
  <dbl>    <dtm>    <dtm>
1     1 2015-01-01 2017-06-01
2     2 2015-02-02 2017-06-01
3     3 2015-02-03 2017-06-01
4     4 2015-01-04 2017-06-01
5     5 2015-01-05 2017-06-01
6     6 2015-01-16 2017-06-01
7     7 2015-05-07 2017-06-01
8     8 2015-03-28 2017-06-01
9     9 2015-07-09 2017-06-01
10    10 2015-10-10 2017-06-01
# ... with 90 more rows
```

- select(vacina, 1,2,3)

```
> select(vacina, 1,2,3)
# A tibble: 100 × 3
      id      dtnasc    dtentrev
  <dbl>    <dtm>    <dtm>
1     1 2015-01-01 2017-06-01
2     2 2015-02-02 2017-06-01
3     3 2015-02-03 2017-06-01
4     4 2015-01-04 2017-06-01
5     5 2015-01-05 2017-06-01
6     6 2015-01-16 2017-06-01
7     7 2015-05-07 2017-06-01
8     8 2015-03-28 2017-06-01
9     9 2015-07-09 2017-06-01
10    10 2015-10-10 2017-06-01
# ... with 90 more rows
```

select ()

- `select(vacina, -dtentrev, -idademae, -escmae)`

```
> select(vacina, -dtentrev, -idademae, -escmae)
# A tibble: 100 × 4
   id      dtnasc vacinado escola
<dbl>   <dtm>   <dbl>   <dbl>
1     1 2015-01-01         1         1
2     2 2015-02-02         0         1
3     3 2015-02-03         1         1
4     4 2015-01-04         0         1
5     5 2015-01-05         1         1
6     6 2015-01-16         1         1
7     7 2015-05-07         1         1
8     8 2015-03-28         0         1
9     9 2015-07-09         0         1
10    10 2015-10-10         0         1
# ... with 90 more rows
```

- `select(vacina, starts_with("va"))`

```
> select(vacina, starts_with("va"))
# A tibble: 100 × 1
  vacinado
  <dbl>
1         1
2         0
3         1
4         0
5         1
6         1
7         1
8         0
9         0
10        0
# ... with 90 more rows
```

select ()

- `select(vacina, ends_with("mae"))`

```
> select(vacina, ends_with("mae"))
# A tibble: 100 × 2
  idademaescmae
  <dbl> <dbl>
1      18      1
2      19      1
3      25      2
4      35      3
5      26      4
6      30      3
7      25      2
8      45      1
9      27      4
10     28      3
# ... with 90 more rows
```

- `select(vacina, contains("esc"))`

```
> select(vacina, contains("esc"))
# A tibble: 100 × 2
  escolaescmae
  <dbl> <dbl>
1      1      1
2      1      1
3      1      2
4      1      3
5      1      4
6      1      3
7      1      2
8      1      1
9      1      4
10     1      3
# ... with 90 more rows
```

rename ()

- rename (vacina, obs = id, vac = vacinado)

```
> rename (vacina, obs = id, vac = vacinado)
# A tibble: 100 x 7
   obs      dtnasc dtentrev vac escola idademaescmae
  <dbl>    <dtm>    <dtm> <dbl>  <dbl>    <dbl>    <dbl>
1     1 2015-01-01 2017-06-01     1     1     18     1
2     2 2015-02-02 2017-06-01     0     1     19     1
3     3 2015-02-03 2017-06-01     1     1     25     2
4     4 2015-01-04 2017-06-01     0     1     35     3
5     5 2015-01-05 2017-06-01     1     1     26     4
6     6 2015-01-16 2017-06-01     1     1     30     3
7     7 2015-05-07 2017-06-01     1     1     25     2
8     8 2015-03-28 2017-06-01     0     1     45     1
9     9 2015-07-09 2017-06-01     0     1     27     4
10    10 2015-10-10 2017-06-01     0     1     28     3
# ... with 90 more rows
```

mutate ()

- mutate (vacina, município = rep(1:5, e = 20), id2 = id+100)

```
> mutate (vacina, município = rep(1:5, e = 20), id2 = id+100)
# A tibble: 100 × 9
   id      dtnasc    dtentrev vacinado escola idade_mae esc_mae município id2
  <dbl>    <dtm>    <dtm>    <dbl>  <dbl>    <dbl>  <dbl>    <int> <dbl>
1     1 2015-01-01 2017-06-01      1      1      18      1      1    101
2     2 2015-02-02 2017-06-01      0      1      19      1      1    102
3     3 2015-02-03 2017-06-01      1      1      25      2      1    103
4     4 2015-01-04 2017-06-01      0      1      35      3      1    104
5     5 2015-01-05 2017-06-01      1      1      26      4      1    105
6     6 2015-01-16 2017-06-01      1      1      30      3      1    106
7     7 2015-05-07 2017-06-01      1      1      25      2      1    107
8     8 2015-03-28 2017-06-01      0      1      45      1      1    108
9     9 2015-07-09 2017-06-01      0      1      27      4      1    109
10    10 2015-10-10 2017-06-01      0      1      28      3      1    110
# ... with 90 more rows
```

mutate ()

- mutate (vacina, idade = dtentrev - dtnasc)

```
> mutate (vacina, idade = dtentrev - dtnasc)
# A tibble: 100 × 8
   id      dtnasc    dtentrev vacinado escola idademaescmaeidade
  <dbl>    <dtm>    <dtm>    <dbl>   <dbl>   <dbl>  <dbl>  <time>
1     1 2015-01-01 2017-06-01         1     1      18     1 882 days
2     2 2015-02-02 2017-06-01         0     1      19     1 850 days
3     3 2015-02-03 2017-06-01         1     1      25     2 849 days
4     4 2015-01-04 2017-06-01         0     1      35     3 879 days
5     5 2015-01-05 2017-06-01         1     1      26     4 878 days
6     6 2015-01-16 2017-06-01         1     1      30     3 867 days
7     7 2015-05-07 2017-06-01         1     1      25     2 756 days
8     8 2015-03-28 2017-06-01         0     1      45     1 796 days
9     9 2015-07-09 2017-06-01         0     1      27     4 693 days
10    10 2015-10-10 2017-06-01         0     1      28     3 600 days
# ... with 90 more rows
```

mutate ()

- vacina2 <- mutate (vacina2, idadeanos = idade / 365)

```
> mutate (vacina2, idadeanos = idade / 365)
# A tibble: 100 x 9
   id      dtnasc  dtentrev vacinado escola idademaescmaeidade idadeanos
   <dbl>    <dtm>    <dtm>    <dbl>  <dbl>   <dbl>  <dbl>    <time>    <time>
1     1 2015-01-01 2017-06-01         1     1      18     1 882 days 2.416438 days
2     2 2015-02-02 2017-06-01         0     1      19     1 850 days 2.328767 days
3     3 2015-02-03 2017-06-01         1     1      25     2 849 days 2.326027 days
4     4 2015-01-04 2017-06-01         0     1      35     3 879 days 2.408219 days
5     5 2015-01-05 2017-06-01         1     1      26     4 878 days 2.405479 days
6     6 2015-01-16 2017-06-01         1     1      30     3 867 days 2.375342 days
7     7 2015-05-07 2017-06-01         1     1      25     2 756 days 2.071233 days
8     8 2015-03-28 2017-06-01         0     1      45     1 796 days 2.180822 days
9     9 2015-07-09 2017-06-01         0     1      27     4 693 days 1.898630 days
10    10 2015-10-10 2017-06-01         0     1      28     3 600 days 1.643836 days
# ... with 90 more rows
```


mutate ()

- vacina2<- mutate (vacina, idade = dtentrev - dtnasc)

```
> mutate (vacina2, idadeanos = idade / 365)
# A tibble: 100 × 9
   id      dtnasc    dtentrev vacinado escola idademaescmaeidade idadeanos
   <dbl>    <dtm>    <dtm>    <dbl>  <dbl>   <dbl>  <dbl>    <time>    <time>
1     1 2015-01-01 2017-06-01         1     1      18     1 882 days 2.416438 days
2     2 2015-02-02 2017-06-01         0     1      19     1 850 days 2.328767 days
3     3 2015-02-03 2017-06-01         1     1      25     2 849 days 2.326027 days
4     4 2015-01-04 2017-06-01         0     1      35     3 879 days 2.408219 days
5     5 2015-01-05 2017-06-01         1     1      26     4 878 days 2.405479 days
6     6 2015-01-16 2017-06-01         1     1      30     3 867 days 2.375342 days
7     7 2015-05-07 2017-06-01         1     1      25     2 756 days 2.071233 days
8     8 2015-03-28 2017-06-01         0     1      45     1 796 days 2.180822 days
9     9 2015-07-09 2017-06-01         0     1      27     4 693 days 1.898630 days
10    10 2015-10-10 2017-06-01         0     1      28     3 600 days 1.643836 days
# ... with 90 more rows
```

`vacina2$idadeanos <- as.numeric (vacina2$idadeanos)`



Utilizando banco de dados nasc.csv

Nascimentos vivos no Brasil por ano segundo idade da mãe.

- Arrumar o nome das variáveis
- Criar variável total_nascimentos (somatório das duas)
- Classificar por ordem de total_nascimentos
- Apagar coluna total
- Criar um novo banco só com variável ano e variável 29 anos e menos