

ME115 - Linguagem R

Turma A - Profa. Tatiana Benaglia

Prova 2/Parte 1 - 08/07/2021 - 2021S1

NOME:João Pedro do Nascimento Sandolin RA:176146

Instruções

- Edite o cabeçalho do arquivo Rmd colocando seu nome e RA.
- Lembre-se de alterar a opção `eval = FALSE` no primeiro chunk para que os chunks subsequentes sejam avaliados.
- Use caminhos relativos para que eu possa executar o seu código no meu computador sem erros e sem ter que editá-lo.
- Para todos os problemas abaixo, escreva o código para responder cada uma das questões.
- Tenha certeza de que o seu arquivo Rmd compila sem erros, gerando um arquivo HTML ou PDF.

Introdução

A Fundação Sistema Estadual de Análise de Dados (SEADE) mantém um painel de dados sobre casos e óbitos relacionados ao coronavírus no Estado de São Paulo a partir de dados oficiais da Secretaria de Estado da Saúde de São Paulo (SES). Um boletim interativo, assim como os dados originais, podem ser acessados em <https://www.seade.gov.br/coronavirus/>.

Você irá trabalhar com uma versão compilada desses dados, contendo informações sobre casos e óbitos por data e municípios do Estado de São Paulo. Os dados estão salvos em um arquivo de nome **covidSP.csv**, disponível no link:

<https://drive.google.com/file/d/1LRm-aQ0zgvjm5PBOKq3N-LhTVcvmv6sU/view?usp=sharing>

Salve-o com esse mesmo no seu diretório de trabalho, ou seja, no mesmo diretório onde você salvou o Rmd do enunciado da prova.

O conjunto de dados possui as seguintes variáveis, mantidas com os mesmos nomes dos dados originais:

- **nome_munic**: Nome do município
- **datahora**: Data no formato YYYY-MM-DD
- **casos_novos**: Casos novos registrados na data
- **obitos_novos**: Óbitos novos registrados na data
- **nome_drs**: Nome do Departamento Regional de Saúde (DRS)
- **pop**: População Estimada (fonte: SEADE)

Leitura dos Dados

Leia os dados no R e salve-o num objeto chamado **covid**, usando uma função apropriada do pacote **readr**. Verifique se os tipos das colunas estão corretos e modifique-os caso seja necessário.

```
library(readr)
covid <- read_csv('dados/covidSP.csv')
head(covid)
```

```
## # A tibble: 6 x 6
##   nome_munic      datahora  casos_novos obitos_novos nome_drs      pop
##   <chr>          <date>        <dbl>        <dbl> <chr>      <dbl>
## 1 Adamantina    2020-02-25         0            0 Marília    33894
## 2 Adolfo        2020-02-25         0            0 São José do Rio~ 3447
## 3 Aguaí         2020-02-25         0            0 São João da Boa~ 35608
## 4 Águas da Prata 2020-02-25         0            0 São João da Boa~ 7797
## 5 Águas de Lindóia 2020-02-25         0            0 Campinas    18374
## 6 Águas de Santa Bár~ 2020-02-25         0            0 Bauru        5931
```

Problemas

Para cada um dos problemas abaixo, use os verbos do pacote `dplyr` para manipulação de dados e o pacote `ggplot2` para os gráficos.

```
library(dplyr)
library(ggplot2)
```

1. [1.0 ponto] Encontre a data em que o primeiro caso de COVID-19 foi registrado em cada município e liste os 10 primeiros municípios. Além disso, quando foi registrado o primeiro caso em Campinas?

```
covid %>% filter(casos_novos > 0) %>%
  group_by(nome_munic) %>%
  slice_min(datahora) %>%
  arrange(datahora) %>%
  head(n=11)
```

```
## # A tibble: 11 x 6
## # Groups:   nome_munic [11]
##   nome_munic      datahora  casos_novos obitos_novos nome_drs      pop
##   <chr>          <date>      <dbl>      <dbl> <chr>      <dbl>
## 1 Ignorado      2020-02-26          1          0 <NA>         0
## 2 São Paulo      2020-02-28          1          0 Grande São Pa~ 1.19e7
## 3 Santana de Parnaí~ 2020-03-07          1          0 Grande São Pa~ 1.38e5
## 4 Ferraz de Vasconc~ 2020-03-12          1          0 Grande São Pa~ 1.93e5
## 5 Carapicuíba      2020-03-14          1          0 Grande São Pa~ 3.95e5
## 6 Mauá            2020-03-16          1          0 Grande São Pa~ 4.60e5
## 7 Santo André      2020-03-16          1          0 Grande São Pa~ 6.94e5
## 8 São Bernardo do C~ 2020-03-16          1          0 Grande São Pa~ 8.12e5
## 9 São Caetano do Sul 2020-03-16          1          0 Grande São Pa~ 1.51e5
## 10 Guarulhos       2020-03-17          1          0 Grande São Pa~ 1.35e6
## 11 Barueri         2020-03-18          1          0 Grande São Pa~ 2.64e5
```

#10 primeiros municípios desconsiderando "ignorado"

```
covid %>% filter(casos_novos > 0) %>% group_by(nome_munic) %>%
  slice_min(datahora) %>% filter(nome_munic == "Campinas") %>%
  select(datahora)
```

```
## # A tibble: 1 x 2
## # Groups:   nome_munic [1]
##   nome_munic datahora
##   <chr>      <date>
## 1 Campinas  2020-03-18
```

#primeiro caso de Campinas

2. [1.0 ponto] Usando a função `epiweek()` do pacote `lubridate`, extraia a semana epidemiológica referente à cada data e crie uma nova coluna nos dados `covid` chamada de `semana_epi` que combina a semana epidemiológica com o ano. Por exemplo, o dia 07-07-2021 corresponde à semana 27/2021, enquanto que o dia 01-07-2020 corresponde à semana 27/2020. Dica: `epiweek()` e `year()` do pacote `lubridate` e `str_c()` do pacote `stringr`.

```
library(lubridate)
library(stringr)
covid %>% mutate(semana_epi = str_c(epiweek(datahora), year(datahora), sep = "/"))
```

```
## # A tibble: 322,354 x 7
```

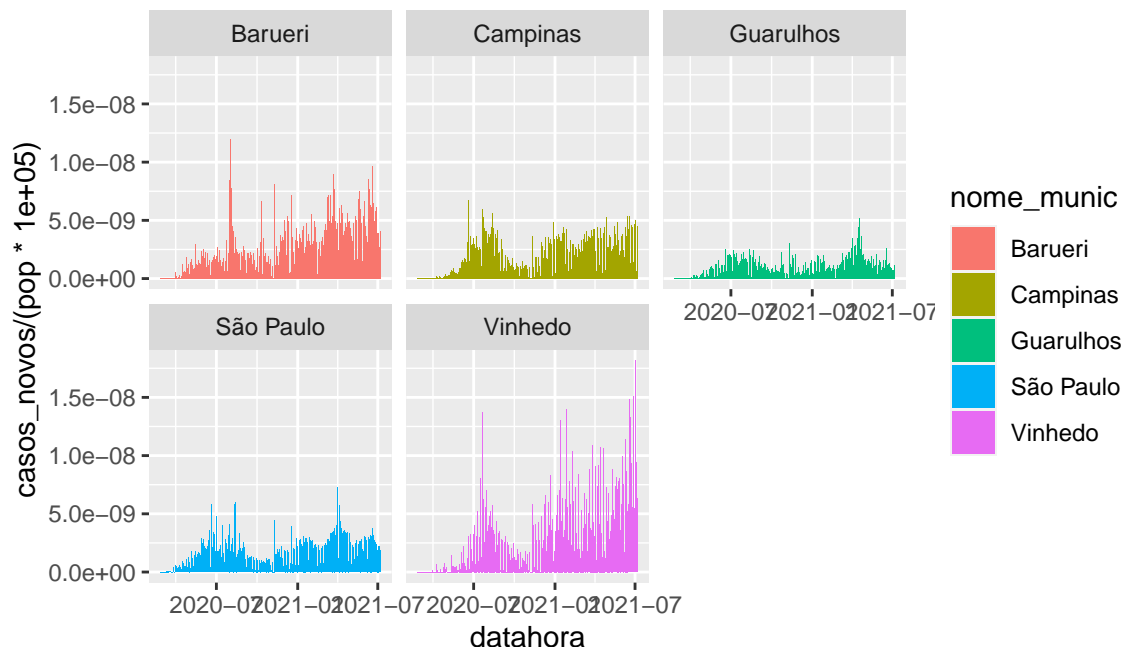
```
##   nome_munic    datahora   casos_novos obitos_novos nome_drs      pop semana_epi
##   <chr>         <date>         <dbl>         <dbl> <chr>      <dbl> <chr>
## 1 Adamantina    2020-02-25           0           0 Marília   33894 9/2020
## 2 Adolfo        2020-02-25           0           0 São José ~ 3447 9/2020
## 3 Aguaí         2020-02-25           0           0 São João ~ 35608 9/2020
## 4 Águas da Pra~ 2020-02-25           0           0 São João ~ 7797 9/2020
## 5 Águas de Lin~ 2020-02-25           0           0 Campinas 18374 9/2020
## 6 Águas de San~ 2020-02-25           0           0 Bauru     5931 9/2020
## 7 Águas de São~ 2020-02-25           0           0 Piracicaba 3122 9/2020
## 8 Agudos        2020-02-25           0           0 Bauru     36134 9/2020
## 9 Alambari      2020-02-25           0           0 Sorocaba  5779 9/2020
## 10 Alfredo Marc~ 2020-02-25           0           0 President~ 3927 9/2020
## # ... with 322,344 more rows
```

3. [1.5 ponto] Selecione três municípios do Estado de São Paulo de sua preferência. Para esses municípios, crie gráficos de barras para representar as seguintes variáveis:

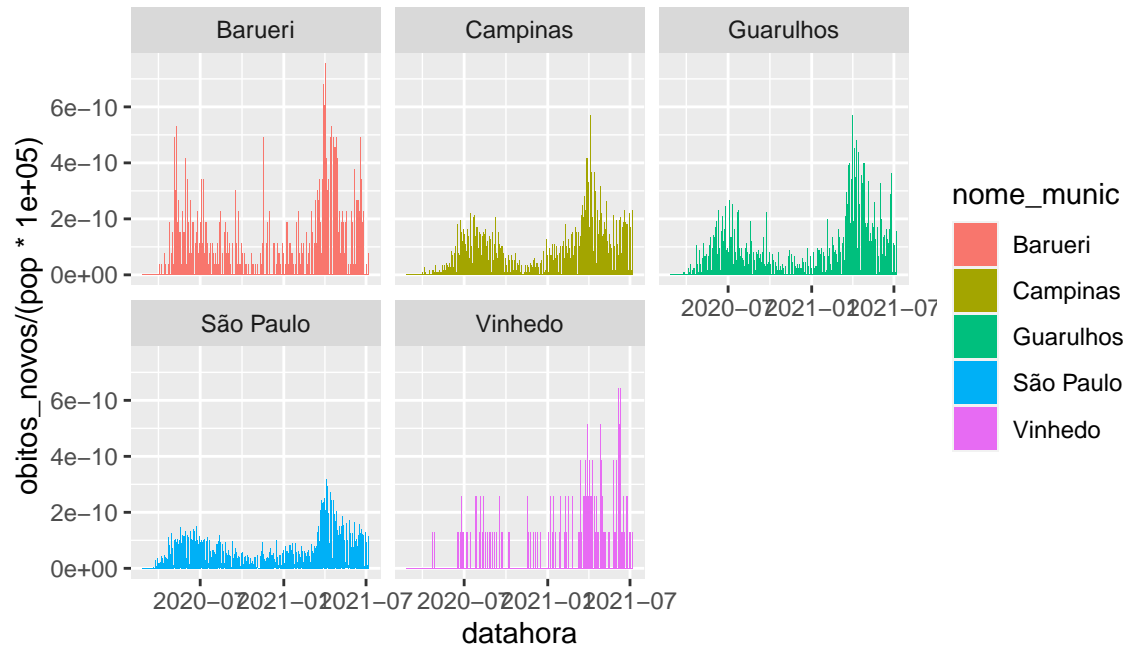
- (a) casos novos/100.000 habitantes por dia
- (b) óbitos novos/100.000 habitantes por dia

Use `facet_wrap()` ou `facet_grid()` para criar um painel de gráficos, um para cada município.

```
#a)
covid %>% filter(nome_munic %in% c("São Paulo","Campinas","Barueri",
                                   ,"Guarulhos","Vinhedo")) %>%
  ggplot(aes (x = datahora, y = casos_novos/(pop*100000)))+
  geom_bar(aes(fill = nome_munic),stat = "identity") +
  facet_wrap(~ nome_munic)
```



```
#b)
covid %>% filter(nome_munic %in% c("São Paulo","Campinas","Barueri",
                                   ,"Guarulhos","Vinhedo")) %>%
  ggplot(aes (x = datahora, y = obitos_novos/(pop*100000)))+
  geom_bar(aes(fill = nome_munic),stat = "identity") +
  facet_wrap(~ nome_munic)
```



O cálculo deveria ser $\text{casos_novos} * 100000 / \text{pop} (-0.2)$

4. [1.0 ponto] Calcule o número total de casos e óbitos acumulados até a data para cada município. Você pode usar a função `cumsum()`. Calcule também a taxa de letalidade, ou seja, a razão entre o total de óbitos e total de casos. Chame essas novas variáveis de `obitos`, `casos` e `letalidade`, respectivamente, e adicione-as nos dados `covid`. Dica: depois de usar `group_by()`, use `ungroup()` no final.

```
covid <- covid %>%
  group_by(nome_munic) %>%
  mutate(casos = cumsum(casos_novos),
         obitos = cumsum(obitos_novos),
         letalidade = obitos/casos) %>%
  ungroup()

head(covid)
```

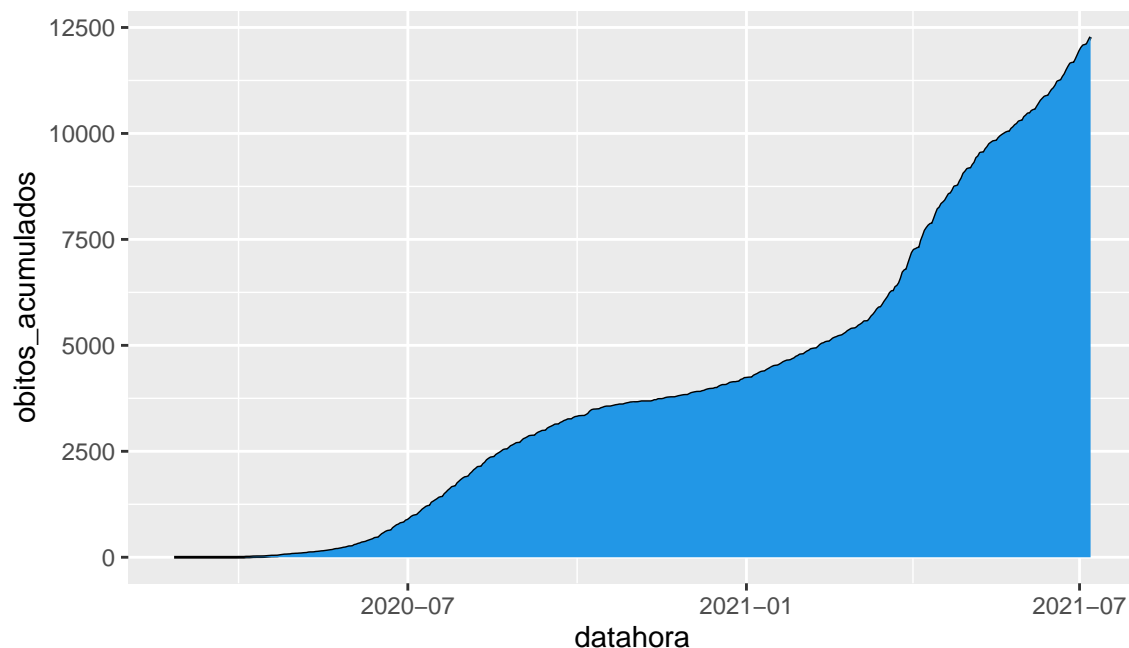
```
## # A tibble: 6 x 9
##   nome_munic datahora   casos_novos obitos_novos nome_drs      pop casos obitos
##   <chr>      <date>         <dbl>         <dbl> <chr>      <dbl> <dbl> <dbl>
## 1 Adamantina 2020-02-25             0             0 Marília   33894     0     0
## 2 Adolfo     2020-02-25             0             0 São José ~ 3447     0     0
## 3 Aguai      2020-02-25             0             0 São João ~ 35608     0     0
## 4 Águas da Pr~ 2020-02-25             0             0 São João ~ 7797     0     0
## 5 Águas de Li~ 2020-02-25             0             0 Campinas 18374     0     0
## 6 Águas de Sa~ 2020-02-25             0             0 Bauru     5931     0     0
## # ... with 1 more variable: letalidade <dbl>
```

5. [1.5 ponto] Considerando apenas o DRS (`nome_drs`) de Campinas, use as variáveis calculadas no item anterior para calcular o número total de casos e óbitos acumulados nesse DRS ao longo do tempo. Faça um gráfico de linha usando `geom_line()` representando o número total de óbitos acumulados até a data. Use também `geom_area()` para preencher a região abaixo da linha.

```
covid %>% filter(nome_drs == "Campinas") %>% group_by(datahora) %>%
  summarise(obitos_acumulados = sum(obitos), casos_acumuladas = sum(casos))
```

```
## # A tibble: 499 x 3
##   datahora    obitos_acumulados casos_acumuladas
##   <date>          <dbl>          <dbl>
## 1 2020-02-25            0            0
## 2 2020-02-26            0            0
## 3 2020-02-27            0            0
## 4 2020-02-28            0            0
## 5 2020-02-29            0            0
## 6 2020-03-01            0            0
## 7 2020-03-02            0            0
## 8 2020-03-03            0            0
## 9 2020-03-04            0            0
## 10 2020-03-05           0            0
## # ... with 489 more rows
```

```
covid %>% filter(nome_drs == "Campinas") %>% group_by(datahora) %>%
  summarise(obitos_acumulados = sum(obitos)) %>%
  ggplot(aes(x = datahora, y = obitos_acumulados)) +
  geom_line(color = 1) +
  geom_area(fill = 4)
```



6. Quais são os DRS (`nome_drs`) com maior número total de óbitos por 100.000 habitantes atualmente?

- [1.0 ponto] Crie um *tibble* chamado `covid_drs` contendo número total de casos e óbitos por 100.000 habitantes na data mais atual (última observação dia 07/07/2021). Imprima esses dados em ordem decrescente de total de óbitos por 100.000 habitantes.
- [1.0 ponto] Faça um gráfico de barras ordenado do número total de óbitos por 100.000 habitantes em 07/07/2021.

```
#a)
covid_drs <- covid %>% filter(datahora == "2021/07/07") %>%
  group_by(nome_drs) %>%
  summarise(casos_drs = sum(casos),
            obitos_drs = sum(obitos), populacao = sum(pop) ) %>%
```

```

mutate(casos = casos_drs/populacao*100000,
       obitos = obitos_drs/populacao*100000) %>%
select(nome_drs, casos, obitos) %>% arrange(desc(obitos)) %>%
na.omit()

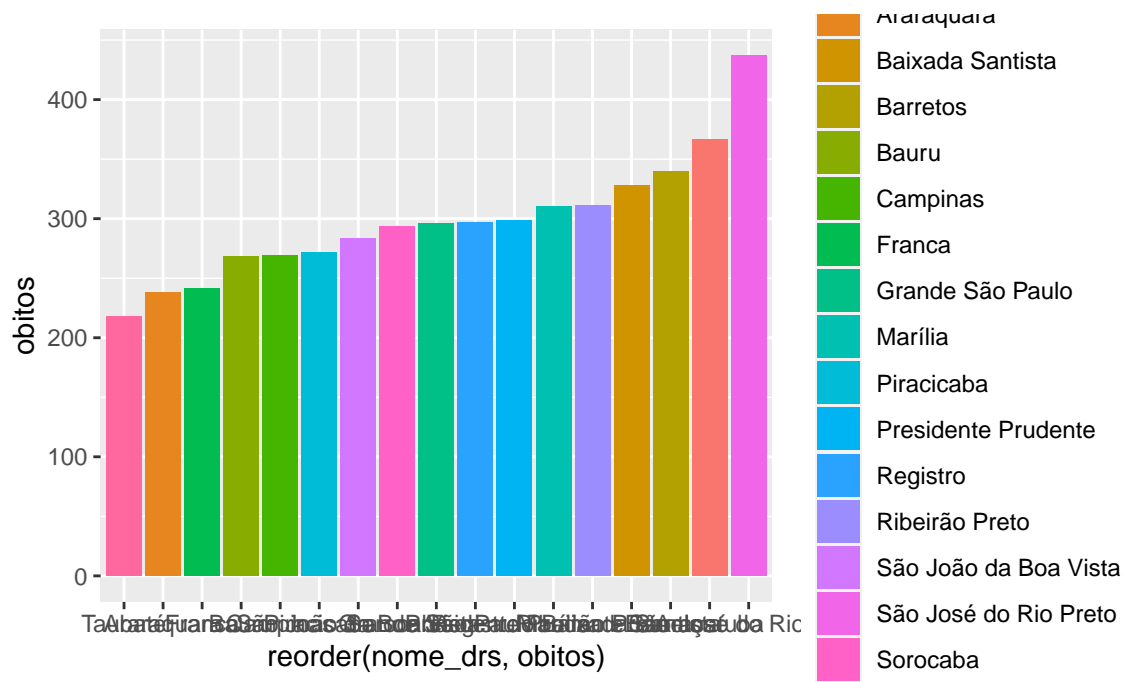
covid_drs

## # A tibble: 17 x 3
##   nome_drs      casos obitos
##   <chr>      <dbl> <dbl>
## 1 São José do Rio Preto 14701.  437.
## 2 Araçatuba      11532.  367.
## 3 Barretos       11227.  339.
## 4 Baixada Santista   7918.  328.
## 5 Ribeirão Preto    9761.  311.
## 6 Marília         10610.  310.
## 7 Presidente Prudente 9739.  298.
## 8 Registro        9718.  297.
## 9 Grande São Paulo   6946.  296.
## 10 Sorocaba        9605.  294.
## 11 São João da Boa Vista 9449.  283.
## 12 Piracicaba     10482.  272.
## 13 Campinas       9222.  269.
## 14 Bauru          10921.  268.
## 15 Franca         8803.  241.
## 16 Araraquara     9979.  238.
## 17 Taubaté        9451.  218.

# O cálculo de casos_drs deveria multiplicar por 100.000 habitantes e não dividir (-0.2)

#b)
covid_drs %>% ggplot(aes(x = reorder(nome_drs,obitos), y = obitos))+
  geom_bar(aes(fill = nome_drs), stat = "identity")

```



Usar coord_flip() melhoraria a visualização

7. Considerando o Estado de São Paulo como um todo:

- [1.0 ponto] calcule a média móvel do número de casos novos dos últimos 7 dias. Dica: função `rollmean()` do pacote `zoo` e cheque os argumentos `k`, `na.pad` e `align`.
- [1.0 ponto] faça um gráfico de barras com o número de casos novos por dia para o Estado de SP. Adicione a esse gráfico uma linha vermelha que representa a média móvel dos últimos 7 dias ao longo do tempo.

#a)

```
library(zoo)
covid %>% group_by(datahora) %>% summarise( media_movel = rollmean(casos_novos,7)) %>%
  arrange(desc(datahora)) %>% head(7)
```

```
## # A tibble: 7 x 2
## # Groups:   datahora [1]
##   datahora   media_movel
##   <date>         <dbl>
## 1 2021-07-07         9.43
## 2 2021-07-07        10.9
## 3 2021-07-07        11.1
## 4 2021-07-07         7.43
## 5 2021-07-07         8.14
## 6 2021-07-07         8
## 7 2021-07-07         8.43
```

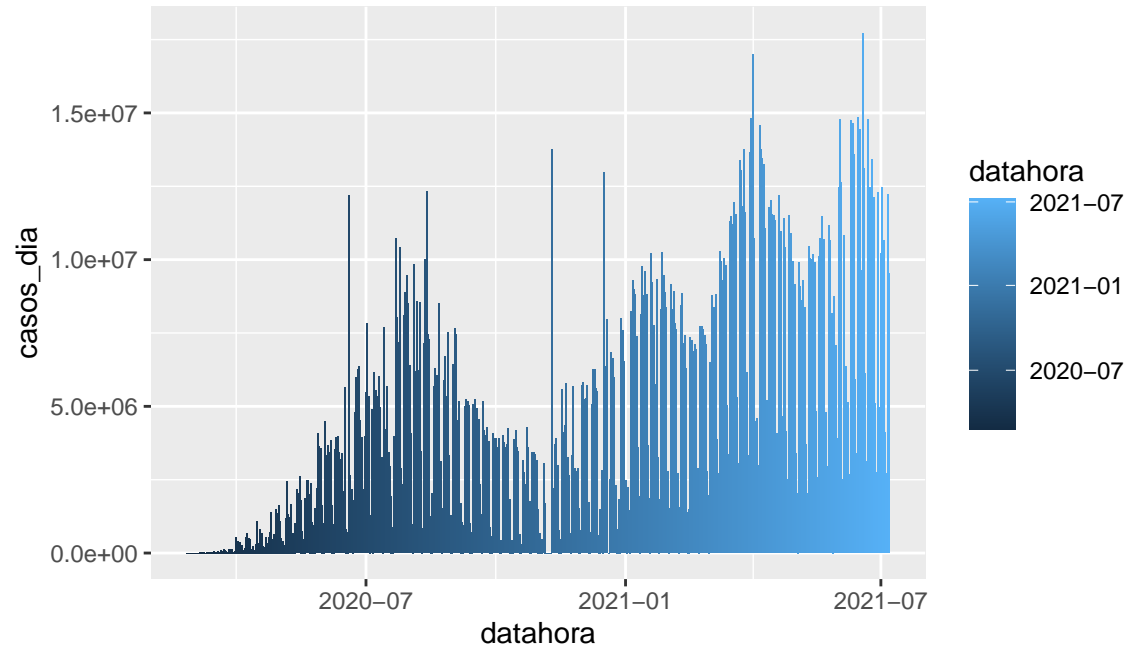
Para cada data, havia a necessidade de calcular o número de casos novos dentro do estado de SP (-0.6)
No uso da função rollmean() faltou inserir o 'align = right' (-0.2)

#b)

```
covid %>% group_by(datahora) %>%
```



```
summarise(casos_dia = sum(casos_novos), media_movel = rollmean(casos_novos, 7)) %>%
  arrange(desc(datahora)) %>%
  ggplot(aes(x = datahora, y = casos_dia)) +
  geom_bar(aes(fill = datahora), stat = "identity")
```



Fez o gráfico de barras corretamente, mas não adicionou a linha com a média móvel (-0.4)